100% AMIGA

Amazing COMPUTING™

Your Original AMIGA™ Monthly Resource

Vol. 3 No. 11 November 1988
U.S. $3.95 CAN $4.95

AMIGA STRUCTURES

More Linked Lists in C

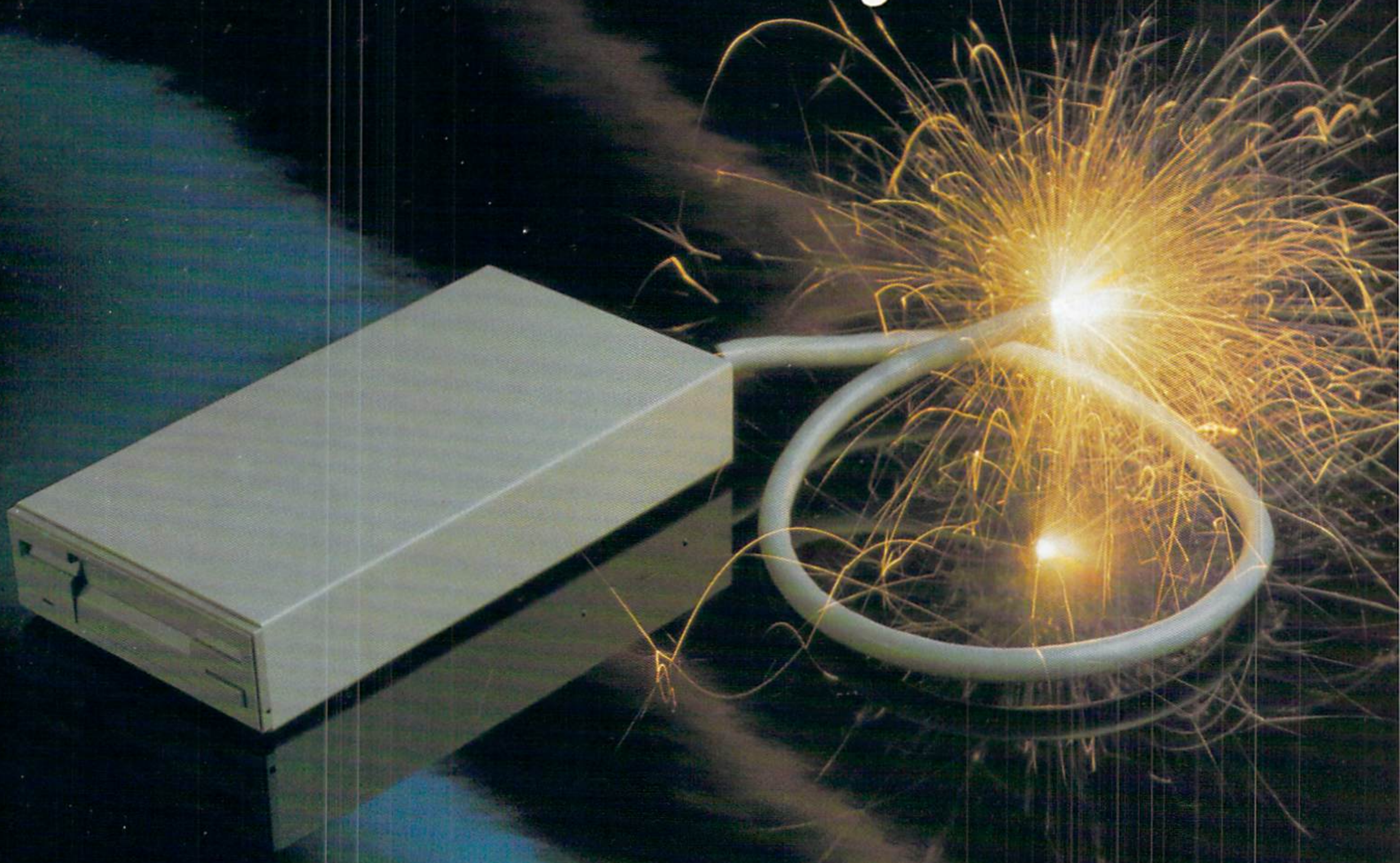
Aegis Modeler: 3D Versatility

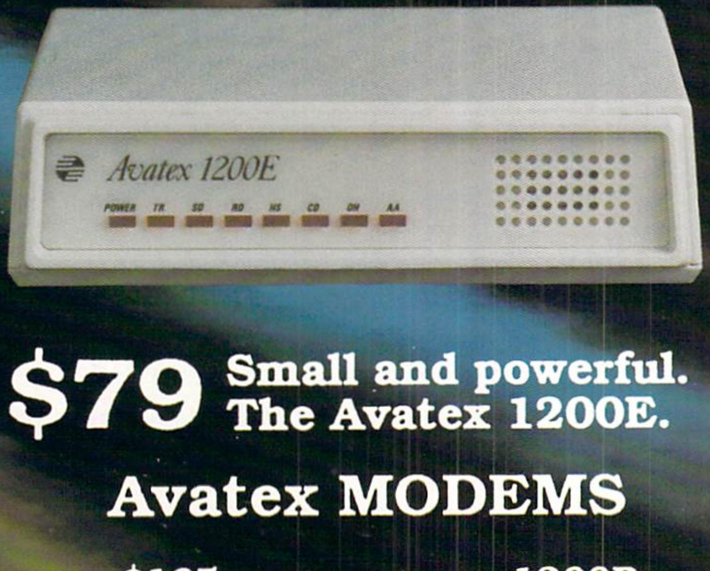
Understanding Structures in C

Adding 3D to your Desktop Videos

AMIGA 2000 Hard Drive Battle

Our DRIVE is Dynamite!





The AIR DRIVE

Dual-Sided, 880K
Color-coordinated chassis and faceplate
Smaller and lighter than the Amiga drive
External drive pass-through
Extra long 30" interface cable
Spring-loaded dust door
Compatible with all Amiga systems
One year warranty
Internal drive also available
CALL FOR DEALER NEAREST YOU.

MEGATRONICS

To Order: 800-232-6342

Ask about our
LOWEST PRICE GUARANTEE

INSIDE UTAH: (801) 752-2642
FAX: (801) 752-8752
CALL FOR A FREE CATALOG.
CREDIT CARDS VERIFIED
FOR YOUR PROTECTION.

MEGATRONICS, INC. BOX 3660, LOGAN, UT 84321

$79 **Small and powerful. The Avatex 1200E.**

Avatex MODEMS

$105	1200P
$99	1200HC
$169	2400
$69	1200i
$159	2400i

FREE WITH EACH MODEM

Amiga communication software
& CompuServe access time.

Starglider II
Dare to Continue the Ultimate Space Fight

The time is the distant future, in a remote corner of the universe. A hostile horde of creatures from the planet Egron is back to capture your home planet, Novenia. You climb into the cockpit to destroy—once and for all—this alien race. Are you ready for the ultimate space fight? Starglider II, the eagerly awaited sequel to Starglider, streaks onto the screen with more astonishing graphics than ever imagined! With a 3D instrument panel and fast, furious, futuristic weapon system . . . it's the most extraordinary cross between flight simulator and shoot 'em up arcade game created!

LANDMARK TECHNOLOGY—ONE STEP FURTHER! Superfast solid 3D graphics give you an incredibly realistic feeling of flight.

SPECTACULAR SOUND EFFECTS add to the game's startling effects.

MULTIPLE CONTROLS, FAST-PACED ACTION with an amazing array of flying creatures and crafts—all aimed at you!

LONG AWAITED SEQUEL to the highly acclaimed Starglider. Now, Starglider II brings you even more technological breakthroughs.

P.O. Box 2227, Menlo Park,
CA 94026, 415/322-0900

RAINBIRD

HOW TO ORDER: Visit your software dealer, or call (800) 227-6900 from U.S. or Canada for Visa, Amex MasterCard, or C.O.D. To order by mail, send check/money order: Rainbird, P.O. Box 8123, San Francisco, CA 94128. CA add 6% sales tax and TX add 7½%. Shipping/handling is $4.50. 2-3 weeks for delivery.

AVAILABLE ON: IBM 256K, CGA/EGA 16 Color $39.95. Commodore Amiga and Atari ST 512K $44.95. Rainbird and Rainbird logo are trademarks of British Telecommunications plc. IBM, Commodore and Atari are registered trademarks of International Business Machines Corp., Commodore Electronics Ltd., and Atari Corp., respectively.

Amazing COMPUTING™

AMAZING COLUMNS

Hot on the Shelves 15
by Michael T. Cabral
Victory over viruses, momentary music, alien annihilation, microfiche mastery, and more!

PD Serendipity 23
by C.W. Flatte
Hot spots from Fred Fish disks 149-152.

Bug Bytes 54
by John Steiner
All the latest from the world of bugs and upgrades.

Roomers 75
by The Bandito
Golden RAM, 16-bit videogames, CD-I, another HAM skirmish ... what could possibly be NeXT?

The Developing Amiga 95
by Steven Pietrowicz
A look at mysteries and successes behind efficient beta testing.

AMAZING REVIEWS

Meet Modeler 3D 19
by David Hopkins
A peek inside a new, open-ended 3D package.

AProDraw Graphics Tablet 21
by Keith Conforti
Artists! Meet the future of Amiga graphics.

StarGlider II 28
by Jeffery Scott Hall
Those irritating Ergons are back for another laser-lashing.

WShell 71
by Lawrence Lichtman
A CLI substitute to spice up your command entry options.

Your Original AMIGA Monthly Resource

• TABLE OF CONTENTS •

Volume 3, Number 11
November 1988

AMAZING FEATURES

Desktop Publishing with ProPage 41
by Barney Schwartz
A no-nonsense tutorial in document creation, plus some jazzy enhancements.

Game Pizzazz! 26
by Jeffery Scott Hall
A grab bag of gaming hints, tips, and high-score crashing secrets.

Structures in C 30
by Paul Castonguay
The backbone of C programming in an understandable nutshell.

On The Crafting of Programs 45
by David Hankins
End the "search" for ways to speed up your programs.

Desktop Video VI: Adding the Third Dimension 49
by Larry White
Unraveling the complexity of 3D for your video creations.

A2000 Hard Drive Round Up 58
by Sheldon Leemon
In the market for an A2000 hard drive? Here's all the information you need.

Keyclick 78
by Mike M. Duppong
Program a typewriter click into your Amiga keyboard.

More Linked Lists in C: Techniques and Applications 81
by Forest W. Arnold
Procedures for managing lists, storing diverse data types in the same list, and putting lists to work in your programs.

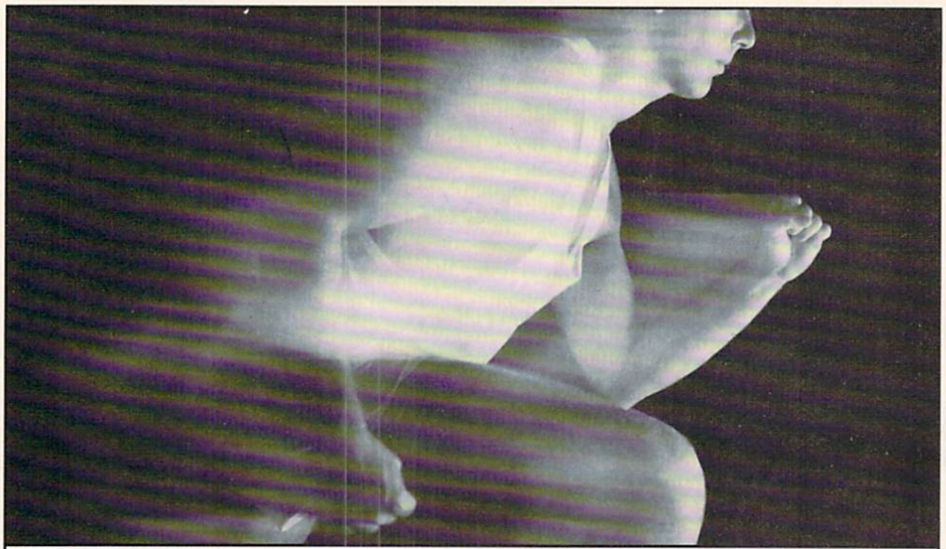
BASIC Linker 97
by Brian Zupke
Combine routines from your program library to create an executable program.

AMAZING DEPARTMENTS

Amazing Mail 8

Index of Advertisers/ Reader Service Card 96

Public Domain Software Catalog 105



In this competitive world there is only one winner.

The results are in and Hurricane has won. According to the August AmigaWorld: "Currently, the Hurricane board is the best way to increase the performance of your Amiga."

Now Ronin has designed a whole family of Hurricane products that will make your Amiga 1000 or 2000 run up to 20 times faster. You'll save hours on ray tracing, desktop video, animation, and other math intensive applications.

The Hurricane family is fully modular, so you can start for as little as $600 and upgrade in stages. Start racing with the 68020. You can then add a 68881. Later move up to our 68030 and maybe the 68882. Add up to 4 megabytes of 32-bit memory and your Amiga will sprint faster than a VAX minicomputer.

Why waste time and money on less performance. Run on in to your favorite dealer and get a Hurricane, or call us for more information. But whatever you do, don't be left at the starting line while everyone else passes you by.

HURRICANE™ *Dealer inquiries invited*

Ronin

P.O. Box 1093, Alameda, CA 94501 U.S.A
Telephone (415) 769-9325
Fax (415) 865-9852 Twx 9102502720 Ronin UD

Amiga is a trademark of Commodore-Amiga, Inc. 68020, 68030, 68881, 68882 are trademarks of Motorola. Hurricane is a trademark of Ronin Research & Development Inc. VAX is a trademark of Digital Equipment Inc.

Amazing COMPUTING™

ADMINISTRATION

Publisher:	Joyce Hicks
Assistant Publisher:	Robert J. Hicks
Circulation Manager:	Doris Gamble
Asst. Circulation:	Traci Desmarais
Asst. Circulation:	Donna Viveiros
Corporate Trainer:	Virginia Terry Hicks
Traffic Manager:	Robert Gamble
International Coordinator:	Marie A. Raymond

EDITORIAL

Managing Editor:	Don Hicks
Co-Editor:	Ernest P.Viveiros Jr.
Co-Editor:	Michael T. Cabral
Hardware Editor:	Ernest P. Viveiros Sr.
Music & Sound Editor:	Richard Rae
Copy Editor:	Michael Creeden

PRODUCTION

Art Director:	Keith Conforti
Illustrator:	Brian Fox
Production Manager:	Rico Conforti
Associate Prod. Mgr:	Mark Thibault

ADVERTISING SALES

Advertising Manager:	John D. Fastino
Marketing Assistant:	Melissa J. Bernier

1-508-678-4200
FAX 1-508-675-6002

SPECIAL THANKS TO:
Buddy Terrell & Byrd Press
Betsy Piper at Tech Plus
Bob at Riverside Art, Ltd.
Swansea One Hour Photo
Boston Jewelry & Loan of Fall River

Amazing Computing™ (**ISSN 0886-9480**) is published monthly by PiM Publications, Inc., Currant Road, P.O. Box 869, Fall River, MA 02722-0869.

Subscriptions in the U.S., 12 issues for **$24.00**; in Canada & Mexico surface, **$36.00**; foreign surface for **$44.00.**

Application to Mail at Second-Class Postage Rates pending at Fall River, MA and additional mailing offices.

POSTMASTER: Send address changes to PiM Publications Inc., P.O. Box 869, Fall River, MA 02722-0869. Printed in the U.S.A. Copyright© Oct 1988 by PiM Publications, Inc. All rights reserved.

First Class or Air Mail rates available upon request. PiM Publications, Inc. maintains the right to refuse any advertising.

Pim Publications Inc. is not obligated to return unsolicited materials. All requested returns must be received with a Self Addressed Stamped Mailer.

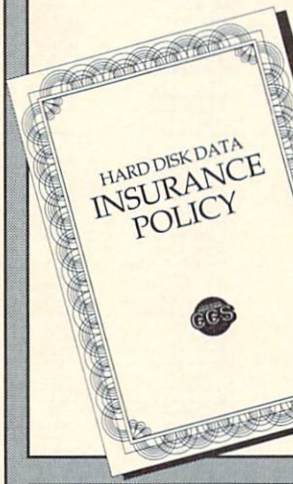
Send article submissions in both manuscript and disk format to the Co-Editor. Requests for Author's Guides should be directed to the address listed above.

FOR THE AMIGA

Protecting your valuable hard disk files is easier and faster than ever before!

QUARTERBACK™

*The **FASTEST** Hard Disk Backup Utility!*



Backup to or restore from:

✓ Floppy Disks ✓ Streaming tape (AmigaDOS-compatible)
✓ CLtd's Konica 10.7MB high-density floppy drive
✓ Inner-Connection's Bernoulli drive ✓ ANY AmigaDOS-compatible devise

✓ Fast backup - 20MB in 30 minutes or less ✓ Uses two floppy drives (if available) for backup/restore with automatic switching ✓ Builds, sorts and displays catalog of files and subdirectories ✓ Provides FULL/Subdirectory/Individual file backup/restore ✓ Includes or excludes files by name (with wild cards), file date, or archive bit ✓ Calculates the number of floppies you'll need before you start ✓ Handles files of unlimited length, unlimited subdirectories and unlimited files per subdirectory ✓ Automatically formats diskettes with no delay as it writes ✓ Sequentially numbers and date/time stamps backup diskettes ✓ Checks the sequence number and date/time stamp of each diskette before restoring files from it ✓ Restores original date/time stamp, file notes, and protection bits on both files and subdirectories ✓ Runs with Workbench or CLI ✓ Produces backup/restore report to disk or printer ✓ Beeps for floppy change ✓ Accepts CLI parameters and batch command files ✓ Detects bad disks during backup or restore ✓ Convenient/user friendly error recovery ✓ Multitasking ✓ Runs in 512K ✓ No copy protection ✓ Works with all AmigaDOS compatible hard disk drives.

Only $69.95 Plus $3.00 shipping and handling, CA residents ass 6% sales tax.

DISK 2 DISK™

Convert C64/C128 Files to the Amiga!

DISK-2-DISK™ makes it easy and convenient to transfer C64/C128 files to and from the Amiga! DISK-2-DISK programs the Amiga model 1020 external 5.25" disk drive to read and write 1541/4040 and 1570/1571 disk formats including 1541 "flippies".

✓ Converts Commodore/PET ASCII to AmigaDOS standard ASCII and vice versa ✓ Transfers word processing text files (such as PaperClip, SpeedScript and Pocket Writer) to and from the Amiga for use with popular Amiga word processors ✓ Includes 3 public domain programs for converting C64 Koala, PrintShop and Doodle files to IFF format ✓ Finds and flags dialect differences between Commodore Basic and Amiga Basic files ✓ Provides VALIDATE BAM and CHECK DISK utilities (VALIDATE BAM verifies the directory structure of the 1541/1571 diskette; CHECK DISK reads every block of a 1541/1571 diskette to detect diskette errors).

DISK-2-DISK requires the Amiga model 1020 5.25" disk drive.

Only $49.95
Plus $3.00 shipping and handling
CA residents add 6% sales tax.

DOS 2 DOS™

DOS-2-DOS transfers MS-DOS and Atari ST files to and from AmigaDOS!

DOS-2-DOS version 3.0 permits access to any MS-DOS volume available via AmigaDOS, including MS-DOS partitions on hard disks and MS-DOS volumes on LANS or SCSI networks.

✓ Supports single and double sided 5.25-inch as well as 3.5-inch 720KB MS-DOS diskettes ✓ Reads/Writes 3.5-inch Atari ST diskettes (GEM format) ✓ Reads a variety of 5.25-inch MS-DOS floppy formats via the CLTD Konica high-density floppy drive ✓ Converts ASCII file line-ending characters and provides Wordstar compatibility ✓ Supports full directory path names, with wild cards in the file names ✓ Allows selection of MS-DOS and AmigaDOS subdirectory and displays sorted directory listing ✓ Formats 3.5-inch and 5.25-inch MS-DOS diskettes and Atari ST diskettes ✓ Provides duplicate file name detection with query/replace options ✓ Provides TYPE and DELETE commands ✓ Permits renaming of files where file name restrictions occur ✓ Remains resident to permit AmigaDOS disk swapping.

Only $55.00
Plus $3.00 shipping and handling
CA residents add 6% sales tax

CCS

Central Coast Software™

268 Bowie Drive, Los Osos, CA 93402 • Telephone: 805 / 528-4906 • FAX: 805 / 528-3138
DEALER INQUIRES WELCOME

MasterCard VISA

Amazing Dealers

The following are Amazing Dealers, dedicated to supporting the Commodore-Amiga™. They carry Amazing Computing™, your resource for information on the Amiga™. If you are not an Amazing Dealer, but would like to become one, call PiM Publications, Inc.:

1-508-678-4200

Alaska	
That Computer Place	Anchorage
Juneau Electronics	Juneau
Alabama	
Abax Data Systems	Huntsville
Alabama Book Store	Tuscaloosa
Command Computers	Birmingham
Madison Book & Computer	Madison
Mel's Photo & Computer Shop	Montgomery
The Computer Nut	Birmingham
The Computer Image	Birmingham
Universal Computer System	Mobile
Arkansas	
Commodore Connection	Sherwood
Microtronix	Fort Smith
The Micro Shop	Little Rock
Arizona	
Books Etc.	Tempe
Books Brothers Ltd.	Tucson
Computer Library, Inc.	Tempe
Computer Works	Glendale
Copperstate Business Systems	Phoenix
North American Digital	Tucson|
Try Tech Inc.	Pheonix
California	
3 E Software And Systems	Hayword
American Mags & Books Ltd.	Englewood
Augnet	Fremont
Balboa Office Supply	San Diego
Bits & Bytes	Fairfield
Bits n Pieces Comp Prod.	Anaheim
Bookland	Marysville
Brown Knows Computing	Redlands
CAL-Tech Bookstore	Pasadena
Candy Computer	Elk Grove
Centerfold News	Los Angeles
Central Park Bookstore	San Mateo
Century Computer Systems	La Habra
Circus of Books	Los Angeles
Circus of Books	Los Angeles
Commodore Land	Alhambra
Commotech	National City
Computer Attic	Palo Alto
Computer Literacy	Santa Clara
Computer Literacy Books	Sunnyvale
Computer Nook, Inc.	San Bernadino
Computer Outlet	San Diego
Computer Spectrum	Burlingame
Computer Time	Citrus Hts
Coronet Newsstand	Oceanside
Covina Bookstore	Covina
Dataphile	Watsonville
Delaurer's News Agency	Oakland
Discount Computer Station Inc.	Long Beach
Empire Computers	Santa Rosa
Encinitas News	Encinitas
ECX Computer Co.	Walnut Creek
Fisher Computers + Software	Rancho Cucamonga
Harding Way News	Stockton
Heinrichson, Schneider & Young, Inc.	Berkeley
Home Computing Center	Concord
Home Computing Center	Mill Valley
Home Computing Center	San Leandro
HT Electronics	Sunnyvale
International Telecine	Santa Ana
K J Computers	Granada Hills
Little Professor Books	Placentia
LTC Computers	Santa Ana
Lyon Enterprises	Berkeley
Micro Flight	San Jacinto
NewsBeat	Davis
Northridge Newsstand	Northridge
Op Amp Tech Books	Los Angeles
Paras Newsstand	San Diego
Put's Electronics	North Highlands
R&R Software	Chula Vista
Ridgecrest Computer Ctr. Inc.	Ridgecrest
Sawyer's News Inc.	Santa Rosa
Software First	Santa Rosa
Software Cellar	Anahiem
Software Central	Pasadena
SoftwareLand	San Diego
Software Plus	Citrus Hieghts
Software Solutions Plus	El Cerito
Stacy's #5	Modesto
Technitronics Service	South Pasadena
The Booksmith	San Francisco
Third Avenue News	Chula Vista
Tower Books	Chico
Tower Books	Citrus Hts.
Tower Books	Concord
Tower Books	Mountain View
Tower Books	Sacremento
Tower Books	San Mateo
The Computer Room	Scotts Valley
Universal News	Hollywood
WCI Computer Central	Eureka
Warner Engineering	San Diego
Winner's Circle	Berkley
World Book & News	Hoolywood
Colorado	
Aurora Newsstand	Aurora
Citadel Computers	Colorado Springs
Computer Discount	Denver
Computer Reports	Boulder
Computer Room	Aurora
Micro Computer World	Fort Collins
Micro World	Lakewood
Sun Country Computers	Colorado Springs
Connecticut	
Aetna Life Club Store	Hartford
Amigo Business Computer	Norwalk
Computer Outlet	PlaVernon
Computer & Software Connections	Plantsville
Derric Electronics	Hamden
Friendly Computers	Plantsville
Mnemonics Inc.	Norwalk
One Byte	Quaker Hill
Personal Computer Center	Norwich
Software City	Orange
Software Kingdom	East Hartford
Tricom Computers	North Windham
District of Columbia	
Programs	Washington
Delaware	
Castle Video	Newark
Newark Newsstand	Newark
Florida	
A A Computers	Jacksonville
Chuck's Newsstand	ViBeach
City Newsstand	Melbourne
Commodore Country Inc.	Pinellas Park
Computer Bar	Pennsacola
Computer Base	Brandon
Computer Image	Miami
Computer Terminal Inc.	Sebring
Computers + Video Creations	Palm Beach
Computers 4 Rent	W. Palm Beach
Computers Plus	S. Daytona
Computer Room	Ft. Lauderdale
Cotronics Inc.	Stewart
Creative Equipment	Miami
Education Computers Etc.	Talahasse
Family Computers	Sarasota
Florida Book Store	Gainesville
Gulf Coast Computer Exchange	Panama City
Heartland Computers	Daytona Beach
Joe's News, Inc.	Miami
Micro Computer Center	Largo
New Age Electronics	St. Petersburg
Newton's Video	Boca Raton
PC Collection	Coral Gabels
Program Store	Clearwater
Prospect News & Books	Ft. Lauderdale
Random Access Computers Inc.	Ft. Walton Beach
RGB Computers and Video Creations	Palm Beach Gardens
Sunny's At Sunrise, Inc.	Sunrise
Computer Room	Ft. Lauderdale
The Open Door	Cocoa
Georgia	
BFJ Enterprises	Roswell
Cardinal Computers	Dalton
Software South Inc.	Savannah
The 64 Store	Atlanta
Hawaii	
Astro Computers & Electronics	Aiea
Computer House Inc.	Honolulu
Computronics	Honolulu
Data-One Computer Center	Honolulu
J & C Repair	Aiea
Idaho	
A.B.I. Computer & Video	Nampa
Boise Computer & Video Inc.	Boise
Illinois	
A Bitcketer	Rockford
Audio Visual Req.	Chicago
Computer Software Connection	Moline
DigitalWorld	Addison
DOS Computer Center	Niles
I.C.V. Data Center Inc.	Mundelein
Keep In Pace Computers	Champaign
M2J2 Inc.	Champaign
Kroch's & Brentano's	Chicago
Kroch's & Brentano's	Schaumburg
Kroch's & Brentano's	Orland Park
Kroch's & Brentano's	Auroa
Micro Age	Oaklawn
Micro Ed Enterprises Inc.	Chicago
Ring Software	Geneva
Software City	Belville
Software Or Systems	Fairview Hts.
Software Plus	Wheeling
Software Plus Chicago	Chicago
Software Plus West	Hanover Park
The Memory Expansion	Bloomington
Unique Computer	Sterling
Warren Radio/FRANCO	Peoria
Indiana	
Bookland	Indianoplis
Burkat Computer Ctr.	South Bend
Bytrex of Angola	Angola
Computer Corner	Marion
Computer People Inc.	Michigan City
Computer Products Unlimited	Ft. Wayne
CPU Inc.	Indianapolis
Indiana News	Indianapolis
Micro Computer Centers Inc.	Indianapolis
Reese Equipment Co.	Plymouth
The Book Corner	Bloomington
The Computer Room of Merrivile	Merrivile
Vons Computers	West Lafayette
Iowa	
Gronet Computers	Des Moine
Micro Computer Applications LTD	Marshaltown
Soft Search	Ames
Kansas	
Arc Electronic Distributors	Wichita
J & L Electronics	Liberal
MidKansas Computers	Newton
Midwest Computers	Manhattan
Software City	Overland Park
Kentucky	
Computer Dimensions Inc.	Lexington
Computer Mania Inc.	Florence
Computer Store of Kentucky	Louisville
Hawley Cooke	Louisville
Reality World Computers	Lexington
Louisiana	
City Newstand	Baton Rouge
Computer Time	Shreveport
Lakeside News	Metairie
Modem Business Machines	Metairie
Software Center International	Metairie
Software Source	Lafayette
The Book Shelf	Slidell
The Computer Clinic	Lafayette
Maine	
Computer Barn	Damariscotta
Hands On Computing	Westbrook
Radio Shack Associate Stores	Jay
Skylight Software	Belfast
Maryland	
Buried Treasure	Rockville
Cal Com Inc.	Silver Spring
Compuvision Vision Center	Columbia
Greetings and Readings	Towson
Knowledgeware Systems	Kensington
Micro Computer Center	Baltimore
Professional Micro Services	Baltimore
Software Advantage	Rockville
Software N Things	College Park
Waldorf Computer	Waldorf
Massachusetts	
Computer Zone	North Attleboro
General Computer Store	Framingham
HCS Computer Center	Marshfield
LCA Video & Computer Ctr.	Norwood
Memory Location	Wellsley
OmniTek Computers	Tewksbury
Pioneer Valley Data Equipment	Amherst
Soft Designs	
Softpro	Burlington
The Bit Bucket	West Newton
The Software Haus	Quincy
Tycom Inc.	Pittsfield
Michigan	
CompuSoft	Kalamazoo
Computer Emporium	Lansing
Direct Access	Lansing
Central Park Computers	Okemos
Global Computer Center	Dearborn
I/O Software	Flint
Michigan News Agency	Kalamazoo
Michigan Software	Novi
Micro World	Livonia
Micro World	Milford
Micro World	Roseville
Micro World	Southfield
Micro World	Southgate
Micro World	Sterling Heights
Microworld Computer Center	Dearborn
Midland Computer Shop	Midland
Norman Business Control Sys.	Wyoming
Readmore Newscenter	Lansing
Rochester Book Center	Rochester Hills
Roseville Computer Store	Warren
Slip Disk	Madison Heights
Software City	Grand Rapids
Specialists Computer	Livomia
State Street Computers	Ann Arbor
T.J. Enterprises	Clinton
The Edge Connector	Clio
United Computer	Canton
Ye Olde Computer Shoppe	Ypsilanti
Video Asis	Sterling Hts.
Minnesota	
Computer Excercise World	Minneapolis
Computer Stuff	Minneapolis
Computers, etc.	Eagan
JMH Software	Maple Grove
Lookahead Software Computer	Northfield
Midwest Computer Services	Clearwater
Shinder's	St. Paul
Shinder's	Maplewood
Shinder's	Burnsville
Shinder's	Minnetonka
Shinder's	Edina
Shinder's	Crystal
Shinder's Read More	Minneapolis
Specialists IN	Hopkins
Valiant Inc.	Stillwater
Mississippi	
Computer Connection	South Haven
National Computer Center	Ocean Springs
Missouri	
Associated Computer Services	Springfield
Brands Mart Computers	Kansas City
Carthage Electronics	Carthadge
Data Byte Computer Co.	St. Louis
Software City	Independence
Systems Plus	Overland Park
Systems Plus of St. Louis	St. Louis
Montana	
Appelgren Computer Systems	Great Falls
Nebraska	
Double E Electronics	Omaha
My Com Inc.	Plattsmouth
Nevada	
Century 23 Inc.	Las Vegas
Computer World	Las Vegas
Tower Records	Las Vegas
New Hampshire	
Dartmouth Bookstore	Hanover
Diversified Computers	Keene
New Jersey	
Business System Professional	Scotch Plains
Family Computer Centres	South Orange
Morris County Stationers	Chester
Software Station	Rockaway
Take One More Byte	Hoboken
The Program Store	Eaton Town
The Program Store	Wayne
Unlimited Software Inc.	Rutherford
New Mexico	
New Horizon Computer System	Alamogordo
Page One Newstand	Albuquerque
Road Runner	Espanola
Technological Concepts	Las Cruces
New York	
Amuse	New York
Amicus- Hudson Valley	Wappingers Falls
Amigo Business Computer	East North Port
Byte Shop	Merrick
Compurite Center	White Plains
Computer Outlet	Jamestown
Computers Etc.	Syracuse
Computerware	East Meadow
Contemporary Computing	Latham
Creative Computers	Johnson City
Cromland Inc.	Brooklyn
Leeder's Video	Schenectady
Leigh's Computers	New York
MCP Associates	Astoria
Mnemonics	North Port
Readers Market	Tonawanda
Software & Such	Scotia
Software City	Forest Hills
Software Link	White Plains
Software Supermarket	Tonawanda
Star Tech Systems	Massena
Synfaire Trading	New York
Sysput Computers	Brooklyn
Tadiran	New York
Tedrow Business Products	Rochester
The Computer Cellar	Albany
Unicom Electronics	Johnson City
Video Computer Bin	Brooklyn
Video Computer Center	Rome
Village Computer Inc.	New York
World Computers Inc.	Hicksville
North Carolina	
Digitz	Raleigh
K&S Newsstand	Clemmons
K&S Newsstand	Kernersville
K&S Newsstand	Winston-Salem
News Center	Cary
Parker News Inc.	High Point
Software City	Raleigh
Software City	Charlotte
TDS Computers	Durham
Triad Computers	Winston-Salem
North Dakota	
Computer Associates	Fargo
G. F. Computers	Grand Forks
Software House	Fargo
The Computer Store	Minot
Ohio	
B5 Software	Columbus
Computer Network	Newark
Computer Site	Cleveland
Earthrise Micro Systems	Columbus
Eddy's Cigar Store	Elyra
Fairborn Home Computer	Dayton
Lakewood International News	Lakewood
Lakes Consumer Electronics	Akron
Leo's Book & Wine Shop	Toledo
Micro Center	Columbus
Microtech Computer Center Inc.	Toledo
Microware Magic	Fairfield
News Shop	Kent
North Coast Programming	Willoughby
Paper & Ink	Tiffin
Softwaire Centre International	North Olmsted
Software and More	Cincinnati
Software Connection	Mayfield Hts.
Specialist Computer System	Toledo
Oklahoma	
Little Professor Books	Stillwater
Merit Computers	Oaklahoma City
Micro Link Computer Store	Tulsa
P C Tech	Tulsa
Second Hand Software	Oklahoma City
Video-Comp Inc.	Lawton
Oregon	
Andromeda	LaGrande
Clackamas Computers	Beaverton
Clackamas Computers	Clackamas
Comm-Shack	Salem
Computron Business Systems	Portland
PC Express	Eugene
Powell's Tech	Portland
Rich Cigar Store	Portland
Sixth & Washington News	Portland
Software Pipline	Beaverton
Tech Books	Portland
Pennsylvania	
Alpha Omega EBE	Warren
ATD Software	Sayre
Basic Computer Systems	Erie
Basic Computer Systems	Hermitage
Circle Computer	Ephrata
Computer Wurx	Stroudsburg
Computers and Games	Reading
Downington Computer Center	Downington
East Coast Software	Hershey
Floppy Disk	Mechanicsburg
Floppy House Software	Palmyra
Gene's Books	King of Prussia
Micro Master Inc.	Monaca
North Central Digital Systems	Danville
Pittsburgh Computer Store	Pittsburgh
Software Hut Inc.	Philadelphia
Tasco Computer Center	Lancaster
Triangle Computers	Indiana
Walden Hill Images	Berlin
Rhode Island	
Micro Limits	Smithfield
Micro Limits	Warwick
Software Connection	Warwick
South Carolina	
Avcom International Inc.	West Columbia
Micro Computer Depot	Sumter
Watson's Jewlery	Chesterfield
Software Haus	Charleston Hts
South Dakota	
Computer Business Center	Rapid City
Tennessee	
ComputerLab of Memphis	Memphis
Computer Hardware Services	Nashville
The 64 Store	Bartlett
Texas	
All Systems Go Software	San Antonio
Amazing Computer Systems	Carrollton
Bookstop	Houston
Bookstop	San Antonio
Byte Shop	Austin
C.E.M. Corporation	Brazoria
Colonel Video	Houston
Computer Age	Houston
Computer Age	Dallas
Computer Magic	Austin
Computer Skills	Euless
FNS Computer Services	Texarkana
Inter Computing Inc.	Grand Prarie
Maxwell Book Inc.	Desota
Metropolitan Computer	Richardson
Micro Search	Houston
Software Library	Witchita Falls
Software Terminal	Fort Worth
The Computer Experience	San Antonio
The Computer Stop!	Amarillo
The Federated Group	Mosquite
Utah	
Magazines and Gifts	Salt Lake City
People's Computers	Provo
Vermont	
Ray Supply Inc.	South Burlington
Virginia	
Family Computer Center	Fairfax
Family Video & Computer Center	Newport News
I/O Computer	Norfolk
Software Central	Norfolk
Software City	Richmond
Tract Software	Prince William
Virginia Micro Systems	Woodbridge
Washington	
Amazing Electronics	Spokane
A.P.P.L.E. CO-OP	Renton
Apple Valley Computers	E. Wenatchee
Bellingham Computer Center	Bellingham
Bits, Bytes and Nibbles	Spokane
Bull Dog News	Seattle
Byte World	Kent
Computer Exchange Services	Bellingham
Computer Quest	Edmonds
Computers + Inc.	Tacoma
Family Computers	Everett
Micro Age Computer Store	Bellevue
Nybbles & Bytes	Tacoma
Omni International	Seattle
Port Book and News	Port Angeles
Programs Plus	Seattle
Real Time Systems	Seattle
Take A Byte	Kennewick
The 68000 Club	Spokane
Tower Books	Bellevue
Y Not Magazines and Books	Federal Way
Wisconsin	
20th Century Books	Madison
Computer Software Center	Milwaukee
Fox Valley Personal Computer	Oshkosh
Madison Programmer's Guide	Madison
Mayfair Computers	Milwaukee
Mom's Computers	Stevens Point
River City Software	La Crosse
Software City	Racine
TMW Software Inc.	Wausau
Wyoming	
Byte Emporium	Laramie
Computer Works	Laramie
Crazy Mountain Electronics	Cooper
Canada	
Compute or Play	Alberta
Computerworks	Alberta
Computer Shop	Alberta
Data Pro	Alberta
Computer Shop of Calgary	Alberta
The Software House	Alberta
Hindsin Computing	Alberta
Red Deer Micro Systems	Alberta
Memory Land	Alberta
Compu Care	British Columbia
Computer King	British Columbia
Conti	British Columbia
Nutec	British Columbia
Pacific Business Technologies	British Columbia
Pacific Business Technologies	British Columbia
Sprta	British Columbia
Strider	British Columbia
Super Software	British Columbia
Canadian Computer Era	Manitoba
Micromart	Manitoba
ComputerWorld	Manitoba
J.W. Allen	Manitoba
Kobetak	Manitoba
Minicomp	Manitoba
Arkon	Ontario
Chatam Computer	Ontario
Computerland	Ontario
Comspec	Ontario
Electronics 2001	Ontario
Hastings Data	Ontario
The Computer & You	Ontario
Thornhill	Ontario
Software Co.	Quebec
Deltronie Solare	Quebec
Merodata	Quebec
Info Mereier	Quebec
Laval Micro	Quebec
Vision Tronique	Quebec
Distribution Visipro	Quebec
Distributed in Canada by	
Ingram Software Distributors	Concord, Ontario
Compulit Distributors	British Columbia
Phase 4 Dist. Inc.	Calgary, Alberta
Argentina	
Logic Computer	Buenos Aires
Australia	
Computermart	Leederville, WA
Diskworks	Greenwich, NSW
John Verhoeven	Cloverdale, WA
Maxwell Office Equipment	Abbotsford, Melburn
Run Soft	Melbourne
France	
A.M.I.E.	Paris
Saudia Arabia	
Abdulla Fouad & Sons	Daman
Scotland	
The Amiga Center of Scotland	Edinborough
Sweden	
Nicholas Peerson	Lund

Amazing Computing™ is also available in most B. Dalton Booksellers, B. Dalton Software Stores, Crown Books, Software Etc., selected WaldenBooks Stores, and Walden's Software Store locations.

Color Graphics.

Multiple Fonts.

Spelling Checker.

Mailmerge.

Multiple Windows.

Resizable Graphics.

Fast Printing.

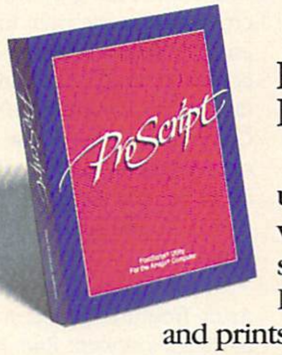
WYSIWYG.

No Other Word Processor For Amiga® Stacks Up To ProWrite™ 2.0.

Word is out. ProWrite 2.0 allows you to expand your Amiga word processing to new horizons. Use it to generate proposals with impressive bar charts. Create letters and reports with snappy graphics. Generate television storyboards. Even dash out newsletters with spot illustrations.

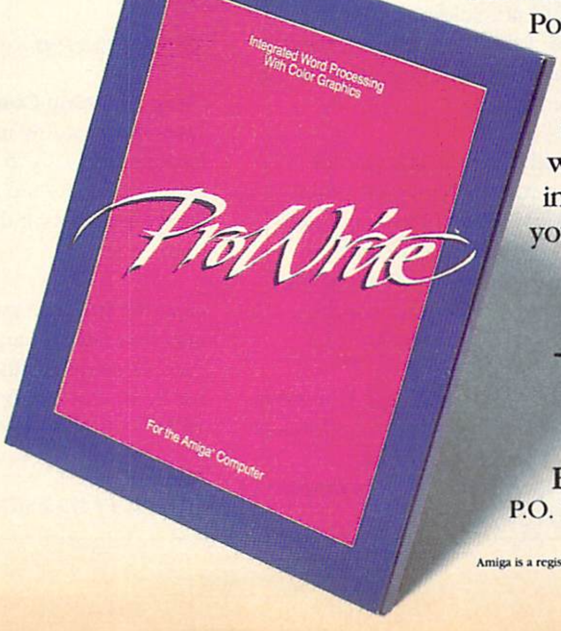
And of course, ProWrite 2.0 has all the other features you expect: a spelling checker with 95,000 word dictionary. Mailmerge. And fast graphics printing. Best of all, it works with our font packages, ProFonts I and II, giving you a broad range of professional and decorative fonts.

All this, plus ProWrite's powerful editing capabilities and ease-of-use combine to make ProWrite the best word processor for the Amiga computer.

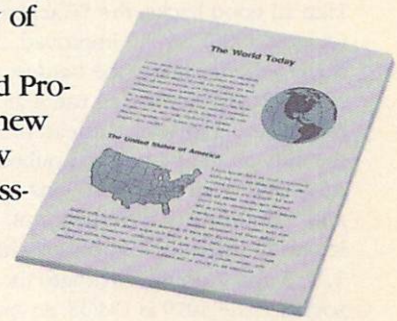


ProScript: Perfection In Print For The Amiga.

ProScript is the new PostScript utility from New Horizons Software that gives you the professional look of typeset copy. ProScript reads ProWrite files and prints them on any PostScript equipped printer, giving you the same multiple font and picture capabilities you expect from ProWrite, but with the typeset quality of PostScript printing.



ProWrite 2.0 and ProScript. A powerful new team that puts a new world of word processing and printing at your fingertips.



New Horizons
SOFTWARE

First in Personal Productivity and Creativity
P.O. Box 43167/Austin, Texas 78745/(512) 328-6650

ProWrite, ProFonts and ProScript are trademarks of New Horizons Software, In
Amiga is a registered trademark of Commodore-Amiga, Inc. PostScript is a registered trademark of Adobe S

AMAZING MAIL

BASIC Bmap Update

Dear Amazing Computing,
Thank you for printing my Basic Bmap suggestion in your September issue. Unfortunately some glitches seem to have crept into the listing. The following lines need fixing before anything will run:

Line #3 should read in part-
```
Q$=CHR$(34)  [not Q$+CHR$ (34)]
```

Line #5 should read in part-
```
Destfile$ [not Destfiles$]
```

Line #8 should read in part-
```
Flength&;":READ  X$ etc. ['"',  not ':']
```

Incidentally, the more adventurous programmers are probably aware that not all of the Bmap file is needed if only a few library functions are being accessed. You take the file apart and keep only the register information that pertains to the specific functions used.

Regards,
Frank Turner
California

Thanks for the follow-up to your own tip. Experience has taught us that this sort of tenacity is the norm for Amiga users. If perfection continues to be the goal of users, the Amiga community will go a long way.

Track Hack

Dear Editor,
Like all good hacks, the "Trackball" (August 1988) can be improved. After buying the trackball (the hardest part), I realized that I wished to use it as a mouse and a joystick. Instead of making the cuts on the pc board, I pulled the ic A5 (4019) from its socket. Pins 1, 3, 5 and 7 of the 4019 were bent out, and the ic was inserted back into its socket. (Pins 1, 3, 5 and 7 do not stick into the socket.) The 4019 is CMOS, so one has to take the standard precautions. Pin 1 was then connected to Point C (Fig. 1) using wire-wrap wire and a low-heat soldering iron. Similarly, pin 3 was connected to Point D, pin 5 to Point A, and pin 7 to Point B. Replacing the cable went according to the instructions in the article except the colors of the wires differed. Now the Atari trackball can be used as an Amiga mouse or joystick at the flip on the built-in switch. I wonder what Jack T. thinks of this?

Yours sincerely,
Wesley
Ebisuzaki
Connecticut

Track Hack II

Dear A.C.
I am writing in reference to the TrackMouse hardware article that appeared in the V3.8 copy of Amazing Computing. There was an error in the wiring instructions on page 36. Mr. Joyce indicated that the right trackball button be connected to the yellow wire of the joystick cable. He also mentioned that the gray wire was not needed, but that it could be connected to the right trackball without ill effect.

In actuality, the gray wire needs to be connected to the right trackball button, and the yellow wire is not needed. The confusion might have arisen from the fact that in the Amiga 1000 User Guide, Commodore lists Mouse Button #2 as pin out #5 (yellow wire) and Mouse Button #3 as pin out #9 (gray wire). The right button on the Amiga mouse is Mouse Button #3!

Aside from this problem, the modification went fine. I would like to tell Mr. Joyce that I (being somewhat of a novice) decided to solder the "trace" connections to the chip socket pins (chip A5), as I could not solder to the traces on the board. I did have to use a meter to determine which socket pins to do the soldering to.

Lastly, I would like to thank you for continuing to run the hardware projects. This is the second one I have completed for my Amiga, and I look forward to more (perhaps a fix that will allow 1000 owners to use the new graphics chip set?). Keep up the good work.

George Houghton
Indiana

Thanks to all for living up to the tradition of true hacking.

Dear Amazing Computing,
I'm just writing to tell you guys what a great job you are doing with your magazine. It is truly one of the best AMIGA magazines along with INFO magazine. In the August issue of your magazine, Graham Kinsey asked anyone who knows the backdoor to ARKANOID to write him. Well, here it is.

1. Pause the game with the space bar.
2. Type "dsimagic" (not the quotes) and press return.
3. Unpause the game by hitting the space bar.
4. If you did everything correctly, a capsule with the letters DS on it will fall from the top of the screen, catch it.
5. Now, you can call up any capsule by hitting its initial.
   ex. L for Laser, C for catch, B for break etc...
   You can also skip screens by hitting the enter key.
   You can go to the final screen by hitting F.
6. In this mode, you can have normally incompatible capsules working.
   ex. You can disrupt the ball and use lasers too.

By the way, I read about this back door from INFO magazine issue #19.

Steve Kee
Canada

For all you frustratred Arkanoid freaks out there, brick-breaking success is only a backdoor away.

Speedy Startup

Dear Amazing Computing,
Over the past few months, I have been looking for ways to speed up my startup-sequence. Here are some time-saving tips I have been using that others might find useful:

-Use the full path name of commands and files rather than searching (set by the PATH command) through the directories for them. For example, instead of:
```
COPY FORMAT TO RAM
```

Use:
```
SYS:C/COPY SYS:SYSTEM/FORMAT TO RAM:
```

A new release from Lattice who brought you the better C:

IT CAME FROM THE C

STARRING LATTICE C++!

- ★ **OBJECT ORIENTED.** A FIRST FOR YOUR OBJECT ORIENTED AMIGA!
- ★ **LIFE** SPRINGS FROM YOUR SCREEN WITH THIS HIGHER LEVEL OF PROGRAMMING!
- ★ **PRODUCE** MORE RELIABLE PROGRAMS WITH FEWER KEYSTROKES & LESS EFFORT!
- ★ **HIDE** YOUR DATA TO PROTECT IT!
- ★ **BUILD** ONE TYPE ON ANOTHER TYPE!
- ★ **DEFINE** A GENERIC THEN REDESIGN SPECIFICS WITHOUT MAKING THE PROGRAM FAIL!
- ★ **TRUE** DATA ABSTRACTION TO AID IN PORTABILITY AND PROGRAM SIMPLICITY!
- ★ **OVERLOAD** OPERATORS AND FUNCTIONS! COMPLEX ARITHMETIC LOOKS LIKE PART OF THE LANGUAGE!

Now at a software supplier near you!

Lattice
Subsidiary of SAS Institute Inc.

Lattice, Incorporated
2500 S. Highland Avenue
Lombard, IL 60148
Phone: 800/533-3577
In Illinois: 312/916-1600

Lattice is a registered trademark of Lattice, Incorporated.
Amiga is a trademark of Commodore-Amiga, Inc.

AT LAST...

An end to the high price of ink-jet ink

inky dink

ink-jet ink for the Xerox 4020, Diablo C-150 and other ink jet printers.

inky dink
gives you the most vibrant colors, the reddest reds, the bluest blues and the blackest blacks.

inky dink
is velvety rich and smooth satisfying the most discriminating palette.

inky dink
comes in three sizes and four colors

inky dink
has been machine and consumer tested for over a year

inky dink
HALF THE PRICE OF THE MANUFACTURERS INK

sensational products from

SOFTWARE SENSATIONS

1441 so. Robertson Blvd.
Los Angeles, Ca 90035
(213)277-8272

XEROX AND DIABLO ARE REGISTERED TRADEMARKS OF XEROX CORP

- When printing multiple lines on the screen, instead of using ECHO, just TYPE an ASCII file, which contains what the ECHO command would have printed out. (i.e. SYS:C/TYPE SYS:S/MESSAGES).

- If a directory, such as SYSTEM, exists on your disk, do not bother to search for it. Just add that directory to the existing path.

- If you have enough memory, use the ADDBUFFERS command. I find that adding 25 buffers to dfO: speeds things up considerably.

- If you wish to execute files from your startup-sequence, make your current directory RAM: before executing the file. This makes the T/Command work file be created in RAM:.

Here is a sample Startup-Sequence:

```
SYS:C/ADDBUFFERS DFO: 25
;adds 25 buffers to dfO:, uses approx. 12.5K
SYS:C/MAKEDIR RAM:C
SYS:C/PATH ADD RAM:C
;Searches RAM:C for commands first.
SYS:C/CD RAM:
; This next line executes a file that copies
; commands to RAM:
SYS:C/EXECUTIVE SYS:S/RAMCOMMANDS
RAM:C/PATH SYS:SYSTEM ADD
RAM:C/TYPE SYS:S/MESSAGES
; Prints out the ASCII file MESSAGES.
```

I hope these tips make booting up your Amiga a little more speedy.

Sincerely,
Mark E. DelCogliano
New York

P.S. I think Amazing Computing is great. I especially like your columns THE COMMAND LINE and C NOTES FROM THE C GROUP, and the technical nature of you magazine. Keep up the good work.

Your speedy startup alterations are appreciated. In your "P.S.," you bring up a point that is always buzzing in the AC editorial corner. Is AC too technical, not technical enough, or somewhere in between? Please let us know what you think. Remember, as readers, you make the editorial decisions.

Galileo Gaffe

Dear AC,
In the introduction to the Amiga Product Guide, Education Edition, you asked for notice of any omissions. Well, I found one glaring omission which I felt I must point out.

Under the science section on page 62, my program, Galileo 2 was not mentioned. Considering Galileo has consistently received glowing reviews, and won title as "most Innovative Educational Program" at the Chicago CES, I was surprised that it was not included. Instead the MicroIllusions program, Planetarium, was listed, even though it has been advertised for a year now, and never shipped. Galileo has been out for nearly 1 1/2 years.

I hope that Amazing Computing will be able to correct this error.

Thank-you for a terrific magazine.

Our apologies to the developers of Galileo and all star-gazing Galileo users. The program is truly a fine indication of how far the Amiga can go in the classroom. You can be sure Galileo will be listed in our Amazing Product Guide: Winter '89. *Thank you for pointing out our gaffe.*

Roomers Rambles

Dear Mr. Hicks:

After reading about Aegis Development, Inc. in your October 1988 "ROOMERS" column, I feel I must write and strongly object to your magazine's continuing improper and potentially libelous editorial policies.

Contrary to your column's remarks, I am principal founder, Chairman, President and Chief Executive Officer of Aegis Development, Inc., and have always been easily located by dialing the correct telephone number (213-392-9972, 9:00-

Other Products from The Other Guys

REASON - a professional proofreading system used by universities and writers around the world to analyze and improve writing. **(Has helped raise students grades when used faithfully.)** $395.00

OMEGA FILE - a *REAL* data base & mail merge $79.99

PROMISE - the *BEST* high speed spell checker. (Even better than Zing!® Spell) $49.99

KEEP-Trak GL - general ledger for home or business $49.99

AMT - amortization program $39.99

MATCH-IT -teaches shapes & colors (preschool) $39.99

MATH-A-MAGICIAN - add, subtract, multiply & divide $39.99

Call or write for more information.

SYNTHIA *High Performance Digital Synthesizer*

A state of the art music tool which will:

Create digital **IFF Instruments** for use with nearly all music programs!

Modifying existing **IFF Instruments.** Use *SYNTHIA* on digitized samples to add reverb, wow, and other enhancements.

SOMETHING FOR EVERYONE:

Additive Synthesis - a traditional method which can create almost any type of instrument.

Plucked String Synthesis - simulates plucked strings . . . right down to the 'pluck'.

Interpolative Synthesis - a method which introduces the natural imperfections found in instruments. (Instruments such as brass, woodwinds, pianos, etc.)

Percussion - build your own drum set . . . create any drum you desire.

Subtractive Synthesis - a simple method of creating instruments.

Special Effects - includes filtering, amplification, phasing, waveshaping, amplitude modulation, real reverb, and . . .

IFF Music Player - powerful and compact. Now you can enjoy those songs that needed a memory expansion before! Up to 32 tracks and 32 IFF Instruments! Supports chords, ties, etc.

NOW SHIPPING

IS IT LIVE . . . OR IS IT SYNTHIA?

Synthia uses the latest technology to generate realistic sounding instruments and even the new families of instruments sound real. A real synthesizer on a real computer!

Why buy digitized instruments when you can *SYNTHIAsize* them?

Requires AMIGA 512K

$99.99

Copyright©1987, THE OTHER GUYS Software • AMIGA is a registered trademark of Commodore Amiga

THE OTHER GUYS

55 North Main Street
Suite 301-D
PO Box H
Logan Utah 84321

(801) 753-7620
(800) 942-9402

VISA MasterCard

5:00 PST). Further, Aegis co-founder Michelle Mehterian is the Corporate Secretary and Vice President of Creative Services. We comprise the original Aegis team and are very much "intact". The original shareholders and executives have, on the other hand, moved on for various reasons. Your other comments regarding the buy out of the company and marketing staff are false and misleading.

Aegis has changed dramatically this year, but for the better. Over the past three months, we have released three new products, *Lights! Camera! Action!*, *Modeler 3D*, and *Aegis Draw 2000*. During the next three months we will release five additional Amiga products (as well as a new product line for the Macintosh II). That will bring our catalog of Amiga products to 20.

I have learned that I am not alone in reading misinformation published by your magazine. I hereby demand an immediate retraction of all false and misleading statements to be published in your next issue.

Sincerely,
David Barrett
President, CEO
Aegis Development, Inc.

As always, if readers have problems or corrections in regard to our editorial content, they need only write us and we will publish their responses. In this vain, we have printed your response in full. To the best of my knowledge no "false or misleading statement" which has been brought to our attention in writing has ever gone unanswered. We continuously publish corrections in the largest format possible because we take our position at AC very seriously. We are a resource that the Amiga community can depend on.

As you are aware, there is a great deal of controversy from the Amiga community on the value of "ROOMERS." From the initial response to our letters column in the October issue of Amazing Computing (including many phone calls), the readers are for it, and the developers are against it. (Although this letter is the only written dissension we have received from a developer.) Our current position on "ROOMERS" remains firm; we will print the column with a disclaimer, and publish conflicting reports and responses as quickly and thouroughly as we are able.

Let me say that the mention of Aegis was small, and that it did end favorably. There was balance to the piece, and a positive statement as to Aegis's position. No attempt was made to discredit your efforts to increase your company's profitability. We are on your side.

Everyone at AC is extremely supportive of any developer who wishes to create products for the Amiga. However, our support must continuously be weighed with the longer view of protecting the Amiga user.

Roomers Rambles

Dear Amazing Computing:
I have been an avid reader of your publication for well over a year. I have always looked forward to reading about new software and hardware being developed for the Amiga. In addition, I have enjoyed reading about new technologies and new markets being opened because of the Amiga. However, the Vol. 3 No. 9 issue seemed to be two-faced.

In one section, I read about an exciting recent AmiExpo show where many new products were shown and announced. Commodore's presence seemed very reassuring. And the large turnout seemed to reflect the increased enthusiasm for the machine. And while I was reading this, I kept thinking to myself, " Terrific! Way to go!"

But a few pages later, I find myself reading a "Roomers" column that seemed intent on chilling the hearts and minds of each and every Amiga owner. Does this Bandito fellow actually believe that Apple has enough clout to, in one swift stroke, halt all of the momentum the Amiga has been gaining over the past three years? No way! If Apple plans on giving the Amiga some serious competition, I say "Fine!" It just may give Commodore more incentive to come out with their new computers and enhancements. Despite this scary column, I see no reason to believe that Commodore will not continue to beat them with their new hardware boasting superior technology at less cost as they have always done.

I can't believe the Bandito really believes the 65816 chip is superior to the 68000. (Gee, maybe if I turbocharge my Commodore 64's 6510 CPU, I can outperform a Mac II...)

My point is to just worry about what is going to happen with the Amiga and forget about looking over our shoulders at Apple. After all, it's what the competition has been doing to Commodore for years now, isn't it?

Sincerely,
Robert E. Wilson
California

Roomers Rambles

Dear AC:
I just read the Roomers column in Vol.3 No. 9 issue.

It looks like Bandito is trying to push a square peg in a round hole. Does he really believe anyone but died in the wool Apple II loyalists will put down $2500 for an 8-bit machine?

The 6502 was a good microprocessor in it's time, but even pushing it to 7 MHz, it's not the wonder chip he makes it out to be.

He fluffs off multi-tasking because of the high costs of memory chips. I think even Commodore doesn't know the convenience and ease of use you get with multi-tasking. I would never buy another computer without it.

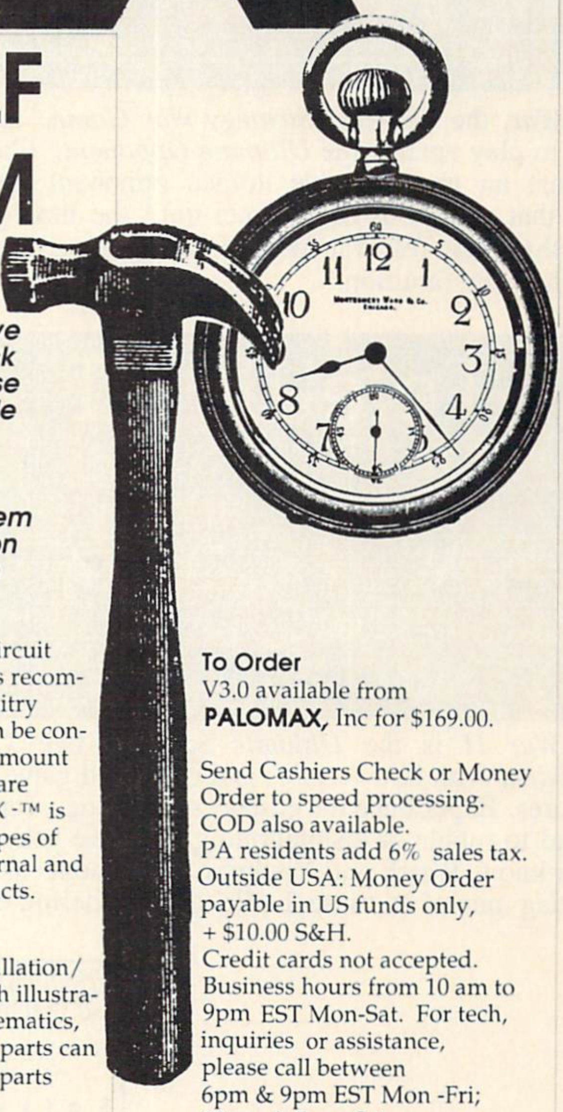
For less than the $2500 system price he quotes for the Apple II GS Plus, I have an Amiga 1000 with 1 3/4 meg of RAM, two 880k floppies and two meg 3 1/2" miniscribe hard drives, RGB monitor and MULTI-TASKING.

Don Morgan
New Jersey

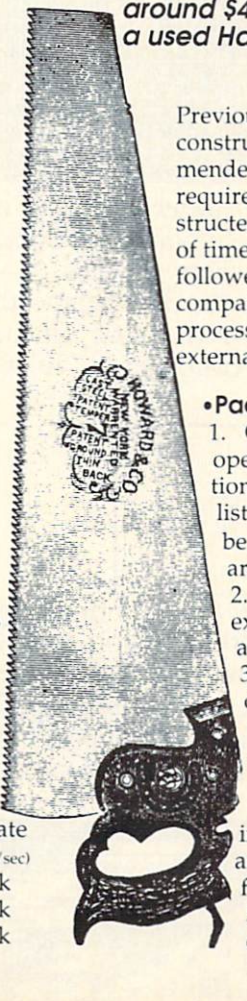
As always, reader feedback about the "Roomers" column is most valuable to us. Since day one, we've been faced with tough questions about "Roomers." Is the column of real value to our readers? Is the column fair to developers? Should we even run it? Again, keep letting us know so we can provide the type of "Roomers" that are fair to everyone.

MAX™

THE BUILD IT YOURSELF HARD DISK SYSTEM



- ***No more painful waiting for icons to come up or large programs to load.***
- ***Supports many 8 and 16 bit controllers, 68000 family CPUs and works with virtually all ST506/412 compatible drives.***
- ***Once constructed, drive configuration is very quick and very easy with the use of a simple ASC11 text file and utilities provided.***
- ***Typical cost for a 16 Megabyte hard disk system around $400.00. (Based on a used Hard Disk of $100)***



Features

1. Customer support (by phone) during construction, installation and operation.
2. Configurable for 1 or 2 controllers with 1 or2 drives/controller.
3. Supports hundreds of megabytes/drive and multiple partitions.
4. Flexible construction configuration: can attach to and pass the 86 pin expansion bus for A500/A1000, or configured for A2000/ZORRO II or ZORRO I 100 pin bus.
5. Text file is used for hard disk configuration parameters
6. Supports SMS OMTI, Adaptec, Western Digital and other controllers.
7. System access rates with V1.3 FastFileSystem and 40ms drives are typically:

	read rate (bytes/sec)	write rate (bytes/sec)
16 bit RLL	524k	291k
8 bit MFM	374k	238k
8 bit RLL	291k	201k

Previous hardware and circuit construction experience is recommended. Minimum circuitry required (3-41Cs) and can be constructed in a reasonable amount of time if all instructions are followed carefully. - MAX -™ is compatible with many types of processors and other internal and external expansion products.

•Package includes:

1. Construction/installation/operation manual with illustrations, layouts and schematics, list of vendors where parts can be purchased, (as no parts are supplied).
2. Device drivers, example configuration and Mountlist files.
3. Sophisticated hard disk exerciser/debugger with fast formatter/verifier, data manipulation and bad track mapping routines.
4. Utility program that includes filecheck, diskcheck and archive bit manipulation functions.
5. Electronic Horizons' comprehensive full and incremental backup and restore utility.

To Order

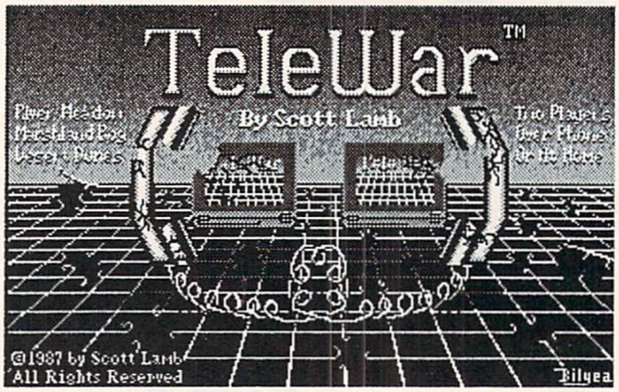
V3.0 available from **PALOMAX,** Inc for $169.00.

Send Cashiers Check or Money Order to speed processing.
COD also available.
PA residents add 6% sales tax.
Outside USA: Money Order payable in US funds only, + $10.00 S&H.
Credit cards not accepted.
Business hours from 10 am to 9pm EST Mon-Sat. For tech, inquiries or assistance, please call between 6pm & 9pm EST Mon -Fri; 10 am & 5 pm Sat.

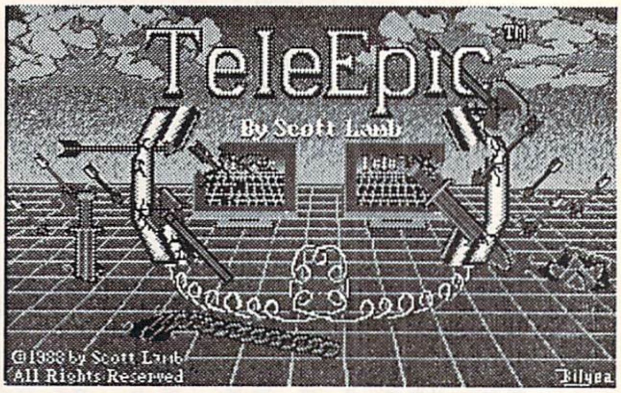
PALOMAX, Inc
424 Moreboro Road
Hatboro, PA 19040
215-672-6815

*- MAX-™ is a trademark of **PALOMAX, Inc.***

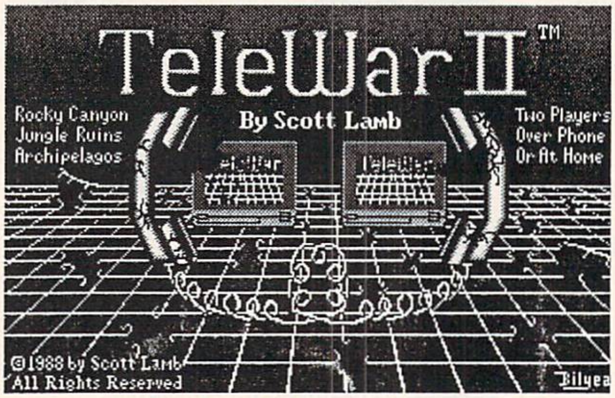
THE ULTIMATE STRATEGY BATTLE SIMULATIONS



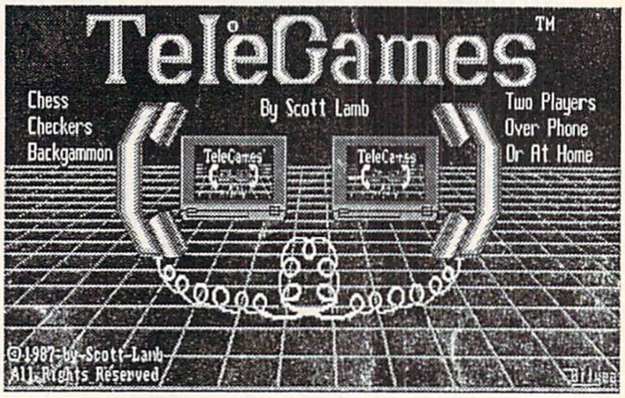
TeleWar, the *Ultimate Strategy War Game*, allows you to play against the *Ultimate Opponent*. Playing against an unpredictable human opponent assures you that victory is only sweet until the next game and that each TeleWar match is an exciting clash of intellect and intuition.



TeleEpic is the *Medieval Version of the Ultimate Srategy War Game*. Instead of commanding tanks and artillery, you lead a band of heroes waging war against your mortal foe. Test your skills wielding a mighty axe or sword. Throw a dagger or check your accuracy as an archer.



TeleWar II is the *Ultimate Strategy War Game Revisited* with new terrains and enhanced game play features. Experience the ecstasy of blasting your best friend to rubble or disintegrating someone you don't even know. Watch out! You may be the next casualty peering out of a twisted pile of smoldering steel.



TeleGames , like all our '*Tele*' products, provides the enjoyment of competition without the problem of where to play. Competition is enhanced with smooth 3D perspective graphic animation and digitized sound effects. Players may compete on one computer or 'TeleConnect' via modem.

INTRUDER
ALERT
Software Terminal

The ***Intruder Alert Monitor*** allows your computer to become a monitoring, logging, or security system for your home or business. The system comes complete with everything you need for remote sensing including detector switches and wire. Intruder Alert Monitor software allows for digitized audio alarms, speech synthesis, modem interaction, date and time stamping of events, external program launching, and utility support for BSR/X-10 controlling. Like all Software Terminal products, direct support from the author and designer is available though our BBS service or via BIX vendor support area 'TeleGames'.

SOFTWARE TERMINAL

Software Terminal 3014 Alta Mere Fort Worth Texas 76116 Phone: (817)-244-4150 BBS: (817)-244-4151

Hot on the Shelves

by Michael T. Cabral

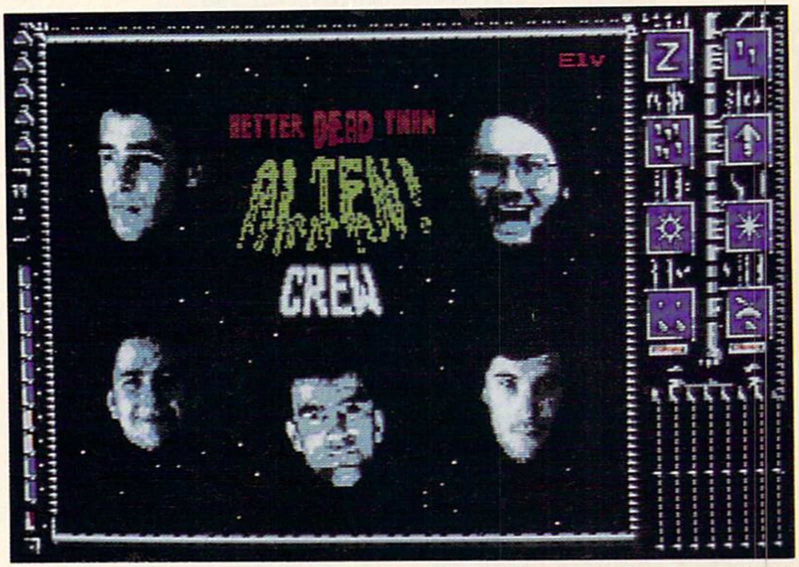
Invading Invaders' Space

Quality lasts. Even in the fickle, short attention-spanned world of video games, the cliche holds true. A few years back, the coin-op classic Space Invaders blasted its way into bars and bowling alleys everywhere. Zapping rows of wiggling aliens became a quarter-gulping addiction and spawned a phenomenon called video gaming. Today's video arcades show how fast that phenomenon has grown up, but Space Invaders clones keep popping up and luring new addicts.

The latest Space Invaders spin-off for the Amiga is the bluntly titled **Better Dead Than Alien**, developed by England's Electra Software and distributed in the U.S. by Discovery Software. Incidentally, Electra is tied to Oxford Digital, crafters of Hunt for Red October and Trivial Pursuit (remember that one?).

The premise, gun down waves of aliens, is familiar, but Better Dead scraps the similarities after that. You step into the moon boots of Brad Zoom, a James Bond-a-like determined to zap all enemies. At risk here is not only the universe, but according to Electra, your lunch as well.

The mutant aliens, "the most revolting forms ever to pollute the universe," are a far cry from the friendly bop-a-longs of the original Space Invaders. In all their grotesqueness, these guys make the unsavory characters of other spin-offs (such as Galaga and Galaxian) look like lady bugs. Bleeeach! And for you wiseacres out there, the floating heads in the photo of the title screen aren't the revolting aliens; they're the notably-better-looking designers of the game!



A peek at some non-dead non-aliens!

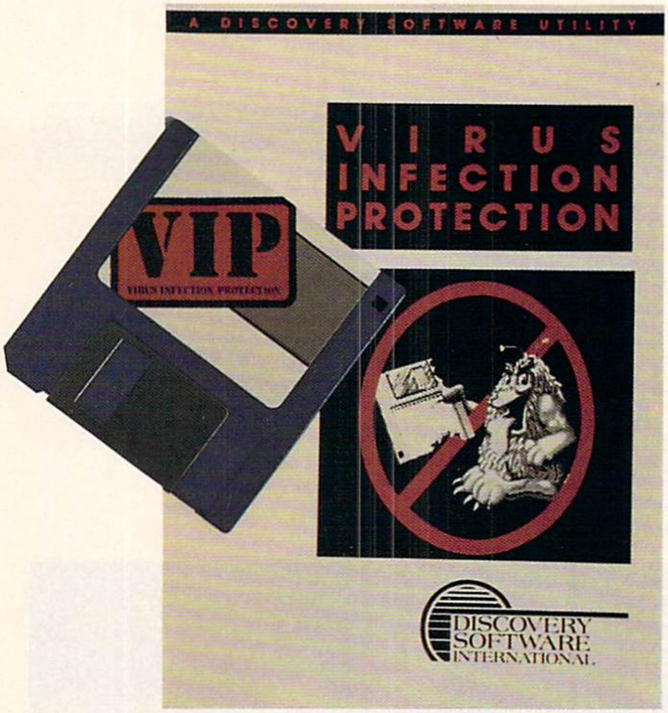
Along with varying degrees of ugliness, the mutants also possess different levels of toughness. Some mutants drop after a single hit, while others remain dangerous under fire. If you can bear to take aim on uglies with glowing green eyes, extra combat power, freedom of movement, and other goodies are in it for you. Lively sound, multi-level action, and a Brad Zoom comic add to the battle. And if you can't stomach it alone, a simultaneous two-player mode is available.

Better Dead Than Alien
$34.95
Discovery Software
163 Conduit Street
Annapolis, MD 21401
800 342-6442

An Amiga Antibody

Better Dead Than Alien lets you expunge unwanted, but thankfully fictitious, visitors. **V.I.P**, another great "Discovery," helps you foil and frustrate the real demons of the computing world: viruses!

If you've been lucky enough to miss the irreparable vandalism of viruses, you may not know just how damaging these things can be. Viruses are invisible programs that latch onto a disk's boot block and wipe out all the crucial information contained in


A DISCOVERY SOFTWARE UTILITY
VIRUS INFECTION PROTECTION
VIP
VIRUS INFECTION PROTECTION
DISCOVERY SOFTWARE INTERNATIONAL

modeler 3D
A 3-D Object Generation System
For the AMIGA® Family of Compute
Modeler 3D™

that block. Worse yet, viruses are more catchy than the common cold. Transfer of the virus from disk to disk is definite and undetectable. A single infected disk can corrupt every bit of information it comes in contact with. Last year's SCA virus told us, "Your Amiga Is Alive!," but this harmless-looking screen message mercilessly translated to "Your Data Is Dead."

With V.I.P., short for Virus Infection Protection, Discovery Software has robbed a favorite credo from the Surgeon General: Prevention is the best medicine. Rather than simply sniffing out viruses, V.I.P. puts your boot block data away for safe keeping long before viruses come slinking along. V.I.P. grabs boots blocks from the files you indicate and packs that information away in a virus-free database. So if a virus slips into your files, all you need to do is get rid of the the virus-damaged files and replace them with your pristine V.I.P. backups.

Like any good doctor, V.I.P also keeps an eye out for viruses that try to sneak in and attack. The Examine command acts as a standard virus checker. So if you suspect you've contracted a bug, V.I.P. can check the suspect block against the safely databased block.

A deficiency inherent to virus checkers, though, is that they can only recognize viruses they've been programmed to nab. What about new viruses—those being diabolically developed right now— even as you innocently type in your valuable bits of data? No problem for V.I.P. If you come across a virus the program does not recognize, you can "teach" V.I.P. to pick out that virus. Just add the virus to V.I.P.'s library, and the next time that virus shows up, V.I.P recognizes it as a varmint. V.I.P. currently cuffs the six most-unwanted viruses: SCA, Byte Bandit, Byte Warrior, Obelisk, North Star, and Revenge.

Discovery Software, themselves players in the international market, also realize that viruses—unlike huge trade deficits, "Freddy" movies, and boring Presidential campaigns—are not exclusively American problems. V.I.P. tosses nationalistic pomp aside and operates in English, German, French, Spanish and Italian.

Viruses, fun bits of prankish programming for some, mean disaster for the rest of us. The problem is big, and getting bigger (even big enough to snag a *Time* magazine cover story in September). Protect yourself now.

V.I.P.—Virus Infection Protection
$49.95
Discovery Software
163 Conduit St.
Annapolis, MD 21401
800-342-6442

Comic Relief

Enough serious stuff. Anybody know the most direct route back to some plain ole' fun? How about a comic book? Everybody reads comics—in the Sunday paper, on the editorial page—just to broad jump over the mental potholes of a day in the real world.

Gold Disk's **Comic Setter** lets you create your own professional quality comics whenever you need a personal chuckle. Comic Setter goes way beyond personal use, too. Rainbowed, comic-set newsletters, advertisements, and other documents are bound to bash boredom, raise eyebrows, and roust results.

The program joins advanced graphic layout tools and standard comic book elements to come up with an easy-to-use fun factory. You can plop any number of any-sized panels on standard-sized pages, along with speech "balloons," cartoonish characters, and comic-style backgrounds. The panels you slap down are automatically masked together and clipped to fit. Multiple magnification settings let work with the full page or just a detailed scrap, and an adjustable grid eases layout pain.

According to Gold Disk, even if you "can't draw," you can create your own characters and backgrounds. If you, like me, are convinced that stick figures are artistic as you'll ever get, you can pick from a packed file of ready-made goons, good guys, and original scenes.

You can also import bitmapped IFF files from other programs.

If you dare to create, you've got 16 colors in interlaced hi-res or square-pixel, medium res in your art bag. Your tools include an airbrush with adjustable "spray distance" and "ink flow," a plentiful mix of line weights and colors, and horizontal and vertical flipping. Rectangles, ellipses, polygons, and Bezier curves are also supported.

To go along with Comic Setter, Gold Disk has also released the first modules in the Comic Art data disk series. Dual-disk sets of Funny Figures, Super Heroes, and Sci-Fi stuff are now available to add some zap, bam, and kaboom to your comics.

Comic Setter *- $99*
Gold Disk
P.O. Box 789, Streetsville
Mississauga, Ontario CANADA L5M 2C2
416-828-7754

Musical ("Post-it") Notes

Creativity always strikes at the weirdest times. As a writer, I've been yanked from sleep, dinner, meetings, and countless other situations by ideas that "just couldn't wait." Diemar Development is out to solve this dilemma for musicians with **MidiVU**, a multi-tasking musical scratch pad for the Amiga.

MidiVU is a desk accesory that records any single track from a MIDI instrument in real time and allows tracks to be saved to disk. The program steals little memory, so it can be tucked away in all your programs for moments of inspiration. If a symphony pops into your head while you are creating an masterpiece in a drawing program or composing a novel with a word processor, MidiVU is at your melodious disposal. Record, Stop, Play, and Loop are the only commands you need.

MidiVU includes a pop-up MIDI Monitor that lets you keep an eye on all messages zipping through the MIDI port. With this option, you can learn the ins and outs of MIDI and catch connection and data problems. The MIDI 1.0 message definitions are also attached to accompanying help screens, and a menu option affords MIDI message filtering.

MidiVU *- $29*
Diemar Development
12814 Landale St.
Studio City, CA 91604-1351
818-762-0804

A 3D Bridge

With the release of **Modeler 3D**, Aegis Development has added some "depth" to the Amiga's trip into the third dimension. This new 3D object generation program lets you create everything from simple spheres and boxes all the way up through pinpoint, CAD-like designs. Modeler, though, does not live by creation alone.

Modeler's biggest plus is its friendly dealings with so many existing software packages. Data from Aegis Draw, Draw Plus, and Draw 2000 can be imported and graduated to 3D with a click of the Exude function. The transformed object, just like any Modeler-created object, is then open to addition of points, polygons, colors, and shading. Sculpt 3D and VideoScape 3D creations can also be imported and deftly edited.

To be fair, the manipulative Modeler is also manipulable. Exports spring to life through Aegis's animation package, VideoScape 3D, and can then be molded as parts of animation sequences. Drawing an on-screen path and setting the number of "tweens" of movement even allows Modeler to take its own camera and object motion files along to Videoscape. With enough memory (1 meg, that is), you can eliminate the in-between step by running VideoScape and Modeler simultaneously. You can also save Modeler objects in Draw Plus or Draw 2000 format for CAD-style plotting in those partner programs.

As a creation tool, Modeler 3D provides top, front, and side view Projection windows for "true to the eye" accuracy. You can also check out objects in solid form for an overall view as you make changes. Multiple layers—eight, in all—let you isolate chunks of complex objects for individual work, and you can zoom in and out for close-up creation. Accuracy is taken even further, to another "dimension" you might say, by a numeric coordinate display and grid snap that allow precise point placement. An on-screen toolbox lets you grab any of these tools anytime.

Menu options simplify tedious operations to mouse clicks. Functions such as Mirror and Array erase useless object duplication steps. Cylinders, planes, boxes, spheres and other 3D primitives are created automatically with menu options. You can shave set-up time by mapping out screen preferences and saving them for later use. Steps necessary for reshaping and assembling primitives into complex objects are neatly taken care of by handy Modify commands. With commands like these, Modeler makes sure you've got plenty of time for the real fun behind 3D!

Modeler 3D
$99.95
Aegis Development
2115 Pico Blvd.
Santa Monica, CA 90405
800-345-9871; in CA 213-392-9972

ALSO NEW & NOTABLE

Two For The 2000

If Commodore's two newest additions to the Amiga 2000 add-on family are any indication, the A2000 is headed out of the living room and into the professional world for good. Aimed at scientific and high-end graphics and animation settings are the **A2090A Hard Disk Controller** and the **A2058 Memory Expansion Card.**

The A2090A controller, updated to include 1.3 autoboot capability, is tagged as the mass storage device you need for full-blown graphic design, animation, and CAD/CAM projects. Both SCSI and ST-506 interfaces are on-board, so you can up to seven SCSI and two ST-506 devices simultaneously for max memory add-on capability. The A2090A also provides buffered direct memory access with high speed burst data transfer.

On the RAM expansion front, a fully-configured A2058 card can pump the A2000 up to 8MB of RAM on a single slot. Two megs are stacked on the standard card with an open door for a whopping six additional megs of RAM in the form of one megabit DRAM chips. The extra RAM goes hand-in-hand with enhanced multi-tasking and a stronger workstation for professional 3D animation and graphic design environments.

Both the A2090A controller and the A2058 RAM expansion card are available from authorized Amiga dealers.

A2090A Hard Disk Controller $399
A2058 Memory Expansion Card $799
Commodore Business Machines
1200 Wilson Drive
West Chester, PA 19380
215-431-9100

DPaint From A Pro

The VCR boom has brought with it the phenomenon of TV tutorials. Jane

Fonda teaches us how to exercise, Omar Sharif teaches us how to play bridge, Dorf (a la Tim Conway) teaches us golf, and Dr. Ruth teaches us...well, you know. Now with the **Video Guide To DeluxePaint II** from Electronic Arts, a professional artist teaches us the finer points of the popular paint program.

A 30-minute, narrated VHS tape lets you use your Amiga to follow the on-screen artist's every move. As he executes a command and creates particular effect, you can see exactly what you need to do to create the same screen image. The visual dimension shows off DPaint II's capabilities and suggests how you can adapt these talents to your graphics needs.

Covered are the creation of a 256 color palette using only 16 basic colors, "greeked" text for preliminary layouts, embossed text and enhanced text effects such as drop shadows and outline. You also learn how to use gradient fills to create landscapes, how to use the Grid tool to craft cityscapes and architectural designs, and how to add depth with the Perspective tool.

Video Guide To DeluxePaint II
$19.95
Electronic Arts
1820 Gateway Drive
San Mateo, CA 94404
800-245-4525

Microfiche Filer: The Sequel

Software Visions Inc., makers of the Amiga's friendliest database, have decided not to rest on the hefty laurels bestowed on them by satisfied users and a wowwed press corps. The recently-released **Microfiche Filer Plus** combines all the features of the original program with a hearty batch of new usefuls.

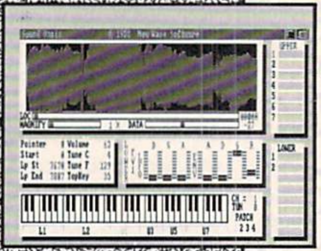
A full AREXX interface, courtesy William Hawes, is the most notable addition. Record selection, form selection, record editing and printing are all macro programmable. The interface also clears the way for complete access to AREXX-compatible word processors, spreadsheets, and paint programs. Now you can do almost anything from within the database—import entire documents into your database, play stored music, run a slide show or animation, telecommunicate, and more.

The brilliantly crafted microfiche metaphor that made the original Filer so easy-to-use is carried through with sophisticated picture handling that includes four squeezing strategies, five automatic color mapping strategies, and pictures in FAST RAM. All IFF, formats—even HAM, brushes, and overscan—are covered. Also new are full field calculation capabilities, including automatic number formatting and field calculation as you type.

If you own the original program, all Microfiche Files files are upwardly compatible. Along the upgrade path, all copies registered by September 30, 1988 can be beefed up for $39, plus shipping and handling. Post-September 30, 1988 registrations require $69, plus p & h. For economy's sake, the old favorite, standard Microfiche Filer will remain available for $99.

***Microfiche Filer Plus** - $179*
Software Visions Inc.
P.O. Box 3319
Framingham, MA 01701
617-875-1238

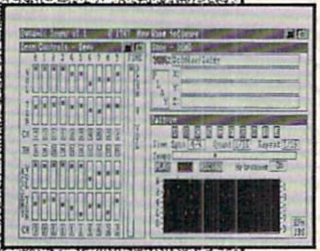
•AC•



Sound Oasis

$99.95

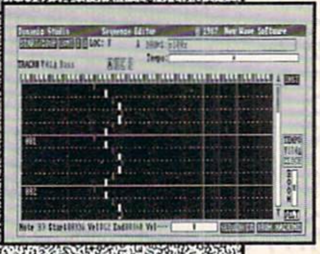
The internal sound capabilities of the Amiga are better than that of any other personal computer. These capabilities mean nothing though, without quality digital sounds, which up till now have been scarce. Sound Oasis gives Amiga owners access to a large library of studio-tested digital samples, by using the Amiga's built in disk drive to read disks made for the Mirage Digital Sampling Keyboard. Sounds can then be played from a MIDI keyboard, the computer keyboard, or saved as an IFF standard file. Mirage is a trademark of Ensoniq Inc.



dynamic drums

$79.95

Transform your Amiga into a professional-quality drum machine with this software package. Easier to use than hardware-based drum machines because everything is displayed graphically on screen. Enter drum patterns quickly and easily in real time with visual feedback and editing. Create realistic drum tracks with any of the 100 drum and percussion samples that are included or use your own unique IFF one-shot samples. Dynamic Drums also has full MIDI implementation and even becomes velocity sensitive when triggered from a MIDI keyboard.



DYNAMIC STUDIO

$199.95

A powerful MIDI sequencer that takes full advantage of the Amiga's sound, graphics, and sophisticated user-interface. Dynamic Studio is perfect for professional applications due to its sophisticated editing capabilities and SMPTE support. It is also ideal for home studios, because in addition to sequencing MIDI instruments, Dynamic Studio has a built-in drum machine, and the ability to playback instruments translated with Sound Oasis.

new wave SOFTWARE

No Copy Protection

P.O. Box 438 St. Clair Shores, MI 48080 (313) 771-4465

Meet Modeler 3D

by David Hopkins

I remember my first look at VideoScape. The program was impressive, to say the least. A 3-D rendering animation package that was easy to operate and produced some incredible animations. It only had one major flaw—objects had to be laboriously mapped out on graph paper and then typed into a text file. I don't know about you, but math isn't one of my better subjects. The last thing I want to do is spend hours over a hot piece of graph paper! And then I heard about...

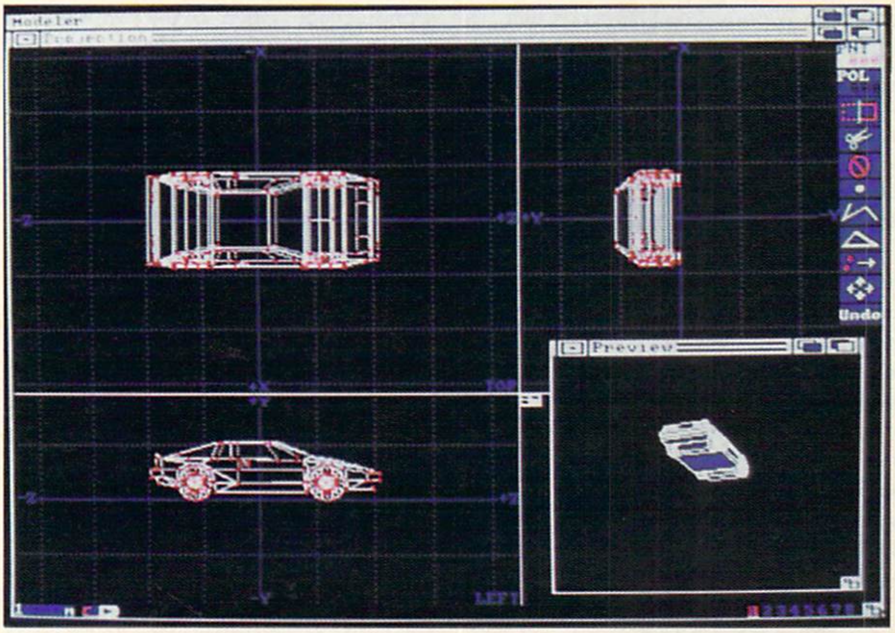
...Modeler 3-D, the little beauty that would allow anybody to sit down at the Amiga and draw out an object. No more typing in endless streams of coordinates. Modeler would do all of that. It would also generate spheres, cones, and all the things VideoScape's EGG and OCT utilities could do. Modeler, though, would add a real user interface *and* an undo! Wow!!

Aegis no doubt grew weary of my constant questioning. Where is it? Where is it? Finally, they gave in. I visited Aegis for a showing of VideoScape 2.0, but they broke down and showed Modeler to me. My whole life flashed before my eyes. It was *wonderful*! I watched as the Aegis rep loaded the F15 jet object from VideoScape, grabbed a wing, and flipped it around. "When?" I asked. "Soon," he answered.

It's shipping!

My hopes dropped when, a couple of weeks later, the same rep said that Modeler had been placed on temporary hold for some major changes. Would Aegis earn a spot in the list of Vaporware Software Announcers?

Well, guess what? Modeler is now reality, shipping as you read this. And I suppose you want to know about it, right? Oh, alright, you talked me into it!



Modeler 3D centers around the standard three view setup.

A New View

Modeler 3D centers around the standard three view setup. The main screen, the "Projection Window," has these three views with a gadget beween them. Dragging that gadget around allows you to rearrange the views so they are as big or small as needed. A bunch of other gadgets surround these views, so let's take a closer look.

Lots of Gadgets

Along the right side of the Projection Window is a whole string of gadgets. PNT displays the number of points selected at any one time. Immediately below PNT is the POL gadget, which divulges that same information about polygons. An edit box allows you to select a piece of an object for further editing (which comes later).

The next gadget, Scissors, cuts things out (you may have guessed). A little "Ghostbuster" symbol below is a sort of "master delete." It deletes anything you select with the edit box; if you do not select anything, it clears the entire page. The next gadget is a curve tool (it looks more like jagged lines) that allows you to draw out connected polygons.

The triangle-shaped tool below uses points you specify to create a polygon. Below that tool is a gadget with two dots and an arrow. After selecting a point or points with the edit gadget, clicking this gadget allows you to point to another place in the 3D view and relocate the point(s) to that place. This provides one way to re-shape an object. A set of arrows on the next gadget allows you to scroll around in each view. Yes, there is an undo...or maybe I should say, an incredible undo gadget. How many programs let you switch between function screens and *still* undo a mistake on the first screen?

The lower left corner of the Projection Window contains a box that shows a scale. Clicking this brings up XYZ coordinates for more accurate placing. Also in the lower left corner is a small padlock, the Lock Plane gadget. A point is created by clicking in two different views. Selecting the same points independently later gets difficult. Locking a plane, for example, lets you say that points from a certain point on are all on the X=0 axis. Simply clicking on the point in one of the other windows selects it. Sounds like a waste, but it comes in *very* handy.

Down in the lower right corner is one of the most useful gadgets—a simple set of the numbers 1 through 8. Each number is a seperate page to work on. Clicking on a number takes you to that page.

What a Selection!

Modeler's menus begin with the usual Project menu containing the Load and Save options. Aegis was kind enough to provide the ability to load not only VideoScape 3D objects, but also Sculpt 3D scenes and drawings created with Aegis Draw Plus. Save allows objects to be saved in standard text mode, new binary mode, or as a motion file. That's right, a camera or object motion can be created by simply drawing it!

The Edit menu has the expected Cut and Paste commands, but it also contains a couple of other handy features. Once an edit box has been placed around an object, the Select option highlights the points or polygons—it's your choice. You can set the polygon color with the Set Color option, a point-and-click chart of colors and attributes (such as matte, glossy, transparent). Polygons can also be flipped or sub-divided here.

The Display menu contains zoom features, as well as a screen selector. Available screens include the already-mentioned Projection window, a 2D screen called a Template, and regular and color previews. The preview modes allow an object to be viewed with it's polygons filled, rather than wire-framed. Rotation around the object is done by clicking around it with the mouse. Modeler can also operate in any of three resolutions: standard interlace screen, a non-interlace screen, or the memory-conserving Workbench screen.

Modeler's Generate menu is basically a reworked version of the old EGG, Easy Geometry Generator. Spheres, planes, boxes, tubes, etc., can be created on any layer. Lathe and Extrude are also available here. The Lathe option tosses you into the Template screen, where the curve tool is used to define a 2D shape. Selecting Do Lathe causes Modeler to ask for the number of rotations and which axis to rotate the object around. *Voila!* A smooth, rounded custom shape is generated!

The Modify menu allows you to rotate, scale, mirror, relocate, etc. and object. Also found here is Remap, which allows an object to be twisted around any axis. Strange effects often result from that one! Finally, an Array option allows duplicate copies of an object to be made on other layers, each offset a specified amount.

Dag'nabbit!

The only real problems I find with Modeler center around personal preference. One quirk is the way polygons are selected. An edit box can be placed around the polygon, or a vertex of the polygon as selected in POL mode. Unfortunately, this also selects a few surrounding polygons with it. Sure, they can be de-selected (kind of a hassle), but it would be much nicer to be able to click on the center of a polygon to select *just* that polygon. A solution might be holding down shift and clicking on other polygons that need the same change or color.

Next, while most of the abilities of EGG and OCT (Object Composition Tool) have been incorporated into Modeler, some important things have been left out. Fractal patches and stars are no longer options; both were handy to have around.

Is it worth it?

Perhaps the best way to illustrate the difference Modeler can make is the following situation. Aegis is preparing another demonstration videotape and some animations to be released on disk, and I was asked to contribute a few.

Within two weeks, I created seven animations that you should be seeing soon. This task would have taken *much* longer if hand-mapping were required. In fact, Allen Hastings, author of VideoScape, tells me that he has switched entirely to using Modeler. This switch from a man who mapped over 500 points for a Lotus automobile by hand!

Over all...

Modeler makes VideoScape an even better package. True, they aren't packaged together, but they are definately a perfect couple. Modeler sells for $99.95 and is an excellent investment for anyone interested in simplifying VideoScape's object creation process.

Modeler 3D
$99.95
Aegis Development
2115 Pico Boulevard
Santa Monica, CA 90405
(800) 345-9871
(213) 392-9972 (in CA)

David Hopkins is now employed in the Creative Services division of Aegis Development Corporation. However, when this article was written, David Hopkins was in no way associated with Aegis or its products.–Ed

•AC•

AProDraw Graphics Tablet

Will it be the Amiga graphics tablet of the future?

by Keith Conforti AC Art Director

The Amiga & The Artist

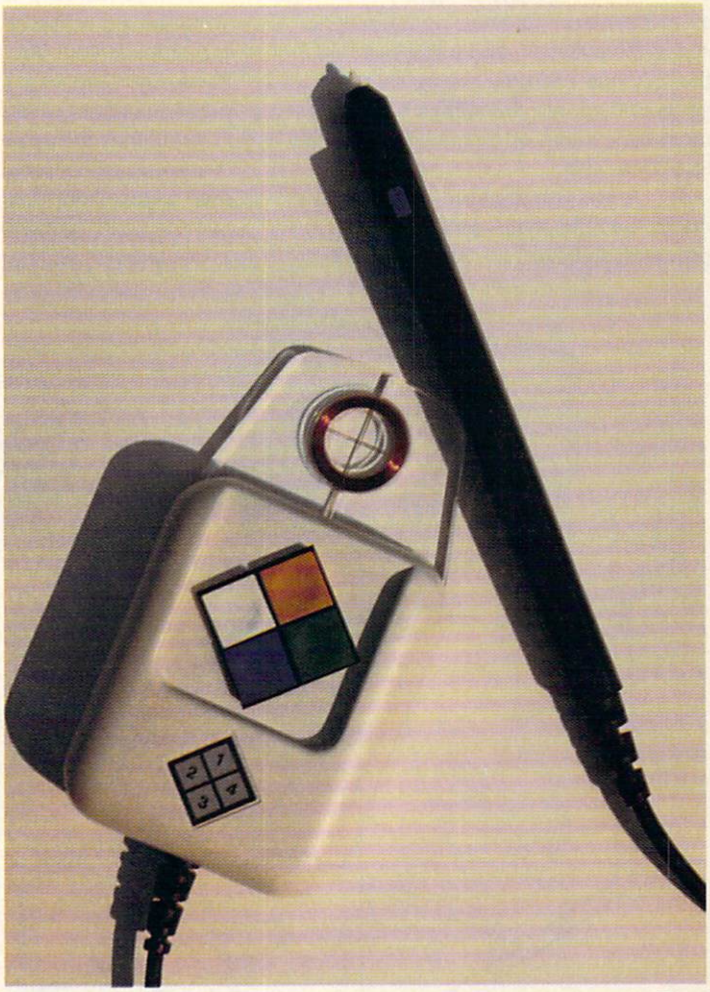
The Amiga is a fantastic graphics machine, so powerful that it could tempt artists to drop pad and palette and pick up pixels and points. The major obstacle preventing the Amiga from becoming the standard in design and graphics are its input devices, all of which have been less than perfect simulations of natural freehand drawing. The devices's main weakness is controlling color and pixel placement. The relative movements of the mouse and the tablet's insensitivity and inaccuracy have discouraged many would-be Amiga artists.

The Amiga & The Artists' Hardware

Software and hardware developers realize the problems in this area and have concentrated much effort into producing better and better input devices for Amiga graphics. One of the latest introductions is a series of graphics tablets developed by SummaGraphics Corporation of Stamford, Connecticut and distributed by R&DL Productions of Long Island City, New York. R&DL currently has two tablets available for the Amiga—AProCAD and AProDraw, the subject of this review.

The AProDraw package includes the tablet, stylus, software tablet driver, a power supply, an optional four-button cursor or puck (for $50), and all the cables you need to get the tablet up and running. It hooks up to your Amiga via the serial port. If you use the tablet on a 1000, you'll need a gender changer. In hooking it up to a 2000, I noticed a slight disturbance the developers overlooked. The connector for the tablet is rather large, and covers one of the audio plugs. This problem is easy to avoid by using only one audio channel. But if you are particular about sound quality, this may annoy you.

The AProDraw tablet is available in two different sizes, 9" x 6" and 16" x 16" (You'll need plenty of desk space for that one!). Each tablet has two application possibilities. It may be used straight, as a versatile graphics tablet on your A500, A1000, or A2000; or if you own a 2000, you can use it as a powerful CAD tablet.



A look at the two drawing devices offered by the AProDraw: the two-button stylus and the four-button cursor.

The tablet driver diskette, produced by R&DL Productions, skirts the mouse signals, allowing your stylus or four-button cursor to control the screen cursor. It's very easy to install and is configured for your Amiga, so there is virtually nothing to do ... Except draw!

What's the Difference Between Left & Right?

AProDraw's most unique feature doesn't even include the tablet. It's the stylus. Quite a few graphics tablets are available for the Amiga, but none address a basic problem : the Amiga's mouse. How do you emulate the right mouse button? The tip of the stylus has always served as the left mouse button on any tablet, but the hardware developers have traditionally forgotten the right button. Well, AProDraw has come to the rescue by putting a "right" mouse button on the barrel of the stylus.

It may not be the most necessary feature of a graphics tablet, but it's nice to see a developer remember that artists use the right button, too!

If you are accustomed to DeluxePaint II, the right mouse button is an important tool for selecting and applying a second color, customizing the palette, sizing the airbrush, and using many of the valuable and time saving shortcuts programmed into the software by that most benevolent of Amiga artist benefactors, Dan Silva.

If you use the four-button puck rather than the stylus, you'll find its precision suitable for CAD work. Why four buttons, you ask? Two function as our familiar mouse buttons. The other two have no function at all on the Amiga. Obviously, this cursor was designed for use with more than one computer configuration. One benefit of using the cursor is that by laying a photograph or another drawing over the tablet, you can digitize right into your CAD program!

Does It Work Well?

AProDraw is a quality graphics tablet. It is versatile and relatively bug-free. Right now it looks like the forerunner in the medium priced graphics tablet market.

The choice of two tablet sizes is a great convenience for artists who are either limited in desk space, or have an abundance of blank drawing table at their disposal.

The AProDraw is extremely accurate and sensitive (so much so that it can emulate your signature almost flawlessly). The only foreseeable problem concerns software. R&DL recommends using the newest versions of most graphics programs. So, if you own DeluxePaint II, you should know that the tablet works best with versions 2.1 or later.

I tested the tablet with DPaint II 2.1 and had no problems whatsoever. R&DL notified me of a bug in a previous version of the tablet driver software. If you press the F10 key while the cursor is positioned over the toolbox on the right side, the program freezes and must be rebooted. The source of this bug was never made clear, but it seemed to be a combination of things concerning their software, the tablet, and DPaint (probably a version older than 2.1). At any rate, R&DL claims everything is now bug free. I tried other programs with AProDraw. Photon Paint, Express Paint, and Digi Paint all worked well and seemed to be compatible with the tablet and software.

Take a Tablet, and Call Me in the Morning

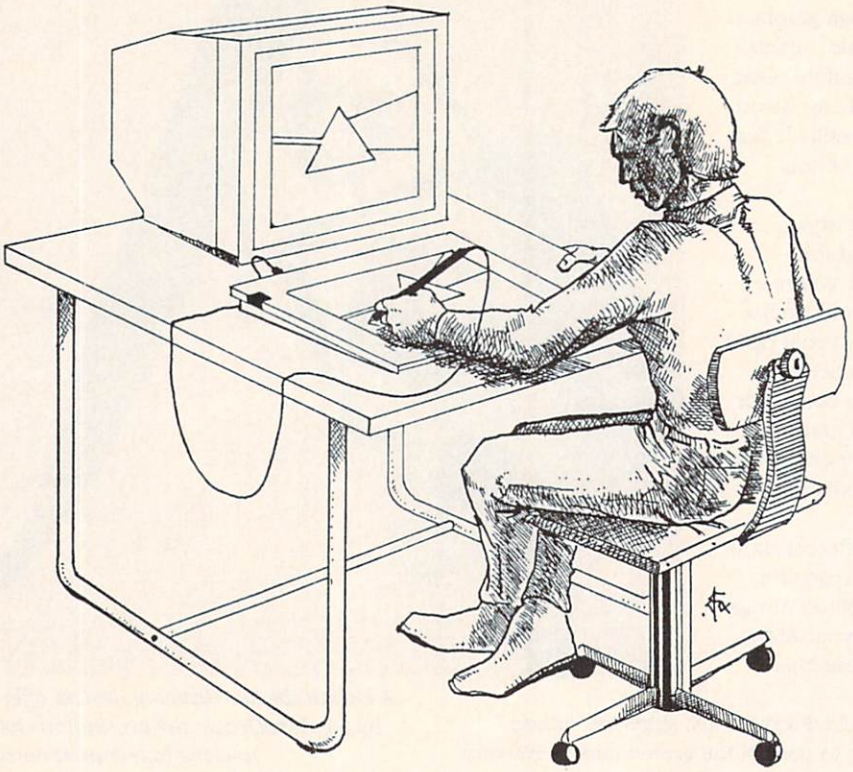
I can rant and rave for ten pages about a graphics tablet and you still may not be impressed with one after you try it. There are three things to consider when you evaluate a tablet.

First, choose a tablet that addresses your artistic needs. You may prefer a tablet with freehand sensitivity and accuracy. You may need a tablet featuring many function "keys"; or you may be looking for something that works well with a CAD package.

Second, choose a tablet that is most compatible with the software you are using. It's not worth buying a tablet that is going to give you some kind of guru every time you use it or one that forces you to go out and spend $500 on new software just to use it.

Third, and most importantly, choose a tablet that fits your price range. You can find a tablet that does everything, but you're going to pay for it. If you choose a tablet that covers your basic needs, functions well with your software, and makes your wallet smile, you've found a tablet that is worth its weight in silicon.

•AC•



AProDraw
9" x 6" tablet $449
16" x 16" tablet $549
Four button cursor $50
distributed by R&DL Productions
11-24 46th Avenue 2A
Long Island City, NY 11101
(718) 392-4090

PD Serendipity

Insigt into the World of Freely Redistributable Software for the Amiga™

by C.W. Flatte

I'm back, this month I'll finish the latest releases from Fred Fish... 150-154. So Let's go!

Fred Fish 150

AirFoil (Update to Fred Fish 71)

Aerodynamics anyone? AirFoil enerates airfoil models as well as their corresponding streamline and pressure distributions. Includes source. AirFoil is by Russell Leighton, with an addendum by David Foster

DC10

Hi! My name is amy...fly me! DC10 is an AmigaBasic DC-10 instrument flight simulator. Appears to be quite in-depth with flight-planning and take-off options along with an extensive documentation file. Note: DC10 requires rebuilding on a separate disk and has been successfully done so by following the author's instructions in the ReadMe_First file. An excellent program by Jan Arkesteijn.

ExecLib

Need a custom library? ExecLib is a nicely working example of how to build and use user-defined disk-resident libraries. Of special interest to developers working with Lattice C. ExecLib is by Alex Livshits

Iconizer

Iconizer is a utility program that saves your current mouse pointer to a small icon. You can restore the pointer just by double-clicking on its icon. Allows for building a whole library of pointers and to use them whenever you want. Binary only. Iconizer is by Alex Livshits

Amazing
COMPUTING
Amiga Shareware & Public Domain Software

Pilot

Pilot is an implementation of the PILOT language for the Amiga, including a demo done for the National Park Service. PILOT is a limited use language for use in educational and computer based instruction programs. Includes the binary only with Beta test kit available from authors. By Terry LaGrone

StealMemBoot

NOFASTRAM? StealMemBoot is a small utility designed to be a direct replacement for NoFastMem kind of programs. It modifies the boot block of a disk, so when you boot with it, all memory allocations will return only CHIP memory. By Alex Livshits

Fred Fish 151

GlobeDemo

Here's one that's outta this world! A graphics demo which displays very smooth transitions of the rotating earth. Features a pop-up menu. Includes source, by Bob Corwin.

Icons

More Icon! Here's yet another potpourri of interesting icons to choose from if you need one for your own program. By Dave Turnock

Pcopy

Make a copy...BACK IT UP! Pcopy is a small intuition-based disk copier similiar to the resident "DiskCopy" except with write-verify and other user-selectable options. Useful for making multiple copies with reliable data. Requires two disk drives. Includes source. Pcopy is by Dirk Reisig.

SCT

SCT is a CLI-based utility (SetColorTable) for displaying and/or setting a screen's colors. You can save the colors of a screen to be restored later, or copy one screen's colors to another. Includes the source. By aklevin.

Amiga Video and Publishing

Complete line of Amiga hardware and software with over 1,200 published titles.

- ✓ Slide Production
- ✓ System Sales
  - Desktop Publishing
  - 2D/3D Video Production
  - CAD Design/Engineering
- ✓ 24-hour Bulletin Board
- ✓ Authorized Amiga Service Center
- ✓ Linotronic 300 PostscriptTypesetting

COMMODORE AMIGA™

Will ship to all 50 states!
24-hour turnaround on in-stock items!
MasterCard/Visa/AmEx phone orders!

METROPOLITAN
Computer Products
800 E. Arapaho Rd. #110
Richardson, TX 75081
(214) 437-9119

SlideShow

Put away that old projectior... SlideShow is a very nicely done slide-show program written in assembly language. Features forward/backward presentation and creative screen wipes. Currently works only with IFF lo-res pictures. Executable only along with some new IFF pictures to have come my way. By Mike McKittrick and Sheldon Templeton.

Surveyor

Where's the pointer? Surveyor is a little utility that opens a window on the current screen and displays information about the pointer. Allows for absolute or relative measurement between two points on the screen. Surveyor is very handy for precise positioning of icons and such. Includes the source. By Dirk Reisig

Fred Fish 152

Blk

Do you have a recursive programming problem? Blk is a requester making tool employing various recursive algorithms including a recursive parser. It takes input text files and converts them to C-source for including as requester declarations. Includes source. By Stuart Ferguson

RunBack

RamBack is a variant of Rob Peck's Run-BackGround program from disk number 73. Allows you to start a new CLI program and run it in the background, then closes the new CLI. This version automatically searches the command-search-path to find the program. Includes source. RamBack is by Daniel Barrett

UUCP

It's here! Finally a version of uucp (Unix to Unix Copy Program) for the Amiga, along with some miscellaneous support utilities like cron, mail, and compress. Includes the source.

Fred Fish153

Dme

More DME! Here's version 1.30 of Matt's text editor. Dme is a simple WYSIWYG editor designed for programmers. It is not a WYSIWYG word processor in the traditional sense. Features include arbitrary key mapping, fast scrolling, title-line statistics multiple windows, and ability to iconify windows. Update to version on disk number 134, includes source. By Matt Dillon

HP11

Can I borrow your calculator? HP11 Emulates an HP11C calculator including the program mode. Features an ON/OFF button that turns the calculator into an icon that will sit and wait until you need it again. Documentation on the features is scarce, perhaps some industrious HP owner could write a small tutorial for the benefit of those that don't own an HP calculator. HP11 is by David Gay

HPMam

A program to manipulate settings and fonts on HP LaserJet+ printers and compatibles. Includes an Intuition interface and some sample picture files. Version 1.0, binary only, shareware. By Steve Robb

Synthemania

Freaky...an interesting, very small (and very persistent!) musical piece. If you plan on stopping it without using three fingers, you better read the document file first! By Holger Lubitz

Fred Fish 154

Ada

Programming in Ada? Ada Syntax checker for the amiga. Includes lex and yacc source. By Herman Fischer and William Loftus.

AssemblyDemos
An interesting group of assembly language demos for your visual and aural pleasure. By Foster Hall

DiskLib
Two utilities for those people who like to split up PD disks into disks of different categories. Includes the source. By Wilson Snyder

Guardian
More medicine for sick machines. Another virus diagnosing and vaccination program. Recognizes any non-standard bootblock. Includes a small utility program to permanently place the program on a copy of your kickstart disk in place of the seldom (if ever!) used Debug() function. By Leonardo Fei

PrintSpool
Spool it with this Print Spooling program. Very useful for printing files in the background. Includes many command-line options. Version 1.0.0, Includes the source. By Frangois Gagnon

Utilities
Lots-O-Utilities! This group of four little utility programs, Cal, Undelete DClock and WhereIs. By Dan Schein

VirusX
An update to the virus-detecting program of the same name on disk number 137. This version also checks for the Byte-Bandit strain. Version 1.21, Includes source. By Steve Tibbett

Virus_Alert!
Yet another anti-virus program with a twist. Once installed a message is displayed just after a warm or cold boot notifying the user that the disk and memory are virus-free, and forcing a mouse-button press before continuing. Anything written to the bootblock thereafter will destroy the message and a normal virus-infected boot (???) will takeplace. Versions 1.01 and 2.01, By Foster Hall.

Wicon
Iconize that window with this "Window Iconifier". Wicon allows you to turn your windows into small icons which can be later recalled. Currently installed with MacWin to give your windows a "rubber- banding" effect. Version 1.14, and includes source. By Steven Sweeting

Gotta go! Until next time...
Gotcha!!!

—C.W. Flatte

If you have any PD Questions, Submissions, old cigars, etc...

Please send them to:

Amazing Computing
c/o C.W. Flatte
P.O. Box 869
Fall River, MA 02720

•AC•

You thought Microfiche Filer was just for *WIMPS**?

Now it's for *PROGRAMMERS*, too!

The easiest-to-use database has just become the most powerful with the introduction of **Microfiche Filer Plus™.**

Microfiche Filer Plus™ includes a full programmer interface using ARexx. Its 33 powerful new commands allow database access from any ARexx macro.

In addition to Microfiche Filer's lightning sort and search facilities, Microfiche Filer Plus™ adds:

- ☐ Automatic Field Calculations
- ☐ HAM Picture Storage
- ☐ Database Programming
- ☐ Direct Control of Other Applications
- ☐ Dynamic Data Import and Export with Other ARexx Compatible Applications

*Window, Icon, and Mouse Persons!

SOFTWARE VISIONS Inc.
P.O. Box 3319, Framingham, MA 01701

Use Microfiche Filer Plus™ to:

- ☐ Control your terminal program and store your downloads
- ☐ Index your disks and execute programs directly
- ☐ Cut and paste data to such programs as TxEd or TeX
- ☐ Store and execute ARexx macros in the database

Only a REAL PROGRAMMER can appreciate the full power of Microfiche Filer Plus™!

MICROFICHE Filer™ PLUS

Only $179. (ARexx sold separately)

For your nearest dealer call 1-800-527-7014.
In MA call (508) 875-1238.

Game Pizzazz!

Hints and tips to move you closer to gaming greatness!

by Jeffery Scott Hall

Thanks to great reader support, I'm back with another article on the hottest, coolest game hints and tips! Game Pizzazz is divided into two sections: Questions and Answers and Reader's Gaming Secrets. Also in this month's article, I have included a special section, Arkanoid Mania. So, dust off your games, get out that joystick, and blast'em!

Questions & Answers

*Q. I am having trouble with the game **F/A-18 Interceptor**. It seems that I can't get game to say I've qualified. After landing on the deck of the carrier at least six times, I still can't obtain qualification for missions. I come to a complete stop on the deck, and I just sit there. Nothing happens! What am I doing wrong?*

Stuart & Donna Wilensky
Wurtsboro, NY

A. A number of things may be going wrong. Instead of guessing, I will give you exact instructions on how to qualify for missions. After taking off from the flight deck, you encounter three enemy fighters directly north. All three fighters must be destroyed before you return. Once this is accomplished, you must land on the ship properly.

Fly past the ship at about 40% thrust and make a U-turn, so you land from the south instead of from the north. The ship is always located at 345 degrees and 145 feet above sea level. As you begin to land, lower the arrestor hook and continue the descent slowly. Just before you land, cut your thrust to 10%. Once on the deck, you see the cables that catch the arrestor hook. Here your plane stops and immediate qualification is given to the pilot.

*Q. When is the proper time to drop the rescue pod in **Interceptor**?*

Anonymous
San Diego area

A. If you are playing the Search and Rescue mission, drop the rescue pod just in front of the floating pilot. Just lower your altitude and cut your thrust until you are skimming the water at a very low speed.

Reader's Gaming Secrets

Faery Tale Adventure

1. An important tip: At the beginning of the game, spend some time in the graveyard south of town. If you are patient, this is also a good spot to get rid of a bone.

2. There are a few places that can be entered, but without sufficient keys, you can be trapped inside forever.

3. As evening approaches, and you are caught out in the open, have one final battle and don't move until dawn. You will be safe where you are standing.

Ultima III

1. In combat with evil monsters, the best way to protect a valuable weapon is to use it.

2. Don't let fear of whirlpools keep you from roaming the high seas.

3. Unless you can stop time, don't wreak havoc in the castle of Lord British. He has a mean temper.

4. Logical positioning of characters is the best defense against invisible monsters.

5. When you reach the machine, recall the words of the Time Lord.

Jim and Maria Marusek
Bloomfield, IN

Arkanoid Mania

Here are several useful game tips that apply to *all* "Arkanoid Imposters." There are also some hints and tips below that are game-specific and are labeled accordingly.

In the higher levels of the "Arkanoid Imposters," certain space walls are indestructible. When this type of wall surrounds all other walls, it is almost impossible to break through. However, this wall does have an Achilles heel. You should notice one brick missing from one of the four sides of that indestructible wall. The trick: get the energy ball through the "hole" in the wall. It's a tight fit, but it goes through if done right.

When too much speed builds up on the energy ball, collect a "slow" power capsule. This takes some of the heat off and returns things to normal.

When playing higher levels, work on the brick walls tactically. Punch a hole through the wall and let the energy ball do all the work, while you collect power capsules and rack up points. For example, knock out the first three or so bricks on the right side of a wall. Now, get the energy ball through that hole you've created. The energy ball has a field day bouncing around inside the

fir.
Julian looked around but discovered nothing.
Julian was hit and killed!
Brv: 35 Lck: 15 Knd: 15 Vit: 1 Wlth: 20

Tips on Faery Tale Adventure should help you avoid being "stabbed in the back"!

walls, and all you have to do is sit back and collect those points and power capsules. But watch out for the return of that energy ball—when the ball escapes the trap, the return is a doozy!

Arkanoid Only!

Want to become invincible, score over one million points, and collect power capsules at will? When you are asked which level you would like to play (or during game play), type DSIMAGIC with no spaces. This opens a "back door" to the game and allows you to command the power capsules to appear at will. Just type "p" for an extra paddle, "b" to break into the next level, etc. If you are at the requester that asks for the start level, you can scroll through levels 1 to 33. To scroll through levels 34 to 66, re-type the password and press F3. Now, you can get even for all those times that you scored a mere 30,000 and got kicked from one end of the galaxy to the other!

Blockbuster Only!

If you want to see the different levels from 1 to 40, enter the following passwords at the title screen's prompt: Gold, Fish. There are two more passwords that reveal levels 50 to 80, but those are for you to figure out.

That's a wrap. Remember game fanatics, this is your forum! Let's hear your game questions, hints, tips, and other information you wish to contribute to the mystical, magical world of computer games. So, until next time, keep those letters coming. Happy gaming!

•AC•

Zap your game letters to:

Jeffery S. Hall
c/o Amazing Computing
P.O. Box 869
Fall River, MA 02722-0869

TO WIN THE AMIGA HARD DISK CONTROLLER GAME...
YOU'D BETTER HAVE A
TRUMPCARD™

NOW SHIPPING!

THE *ACE* OF HARD DISK CONTROLLERS...

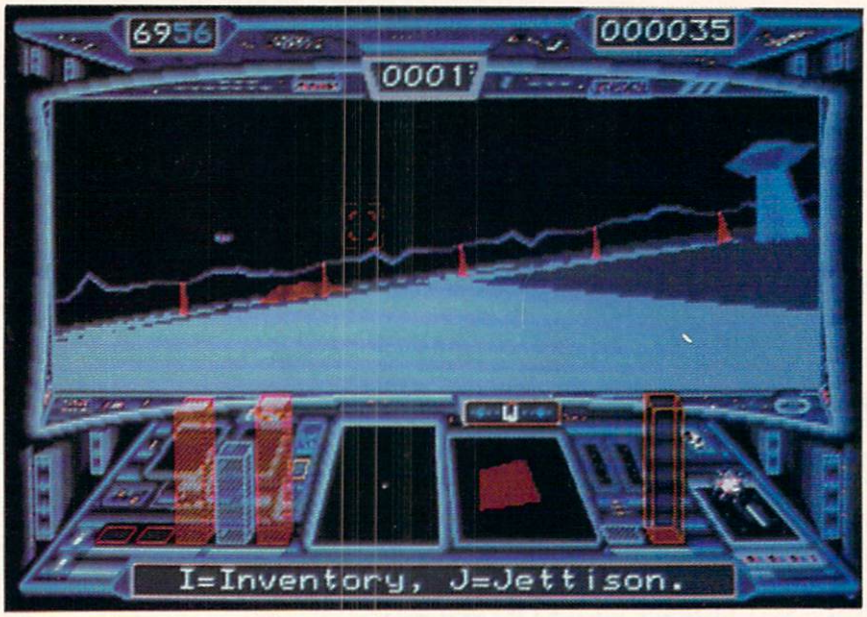
- SCSI INTERFACE SUPPORTS UP TO 7 DRIVES CHAINED TOGETHER
- AUTO BOOT ROM SOCKET IS READY TO GO WITH RELEASE OF 1.3
- SHORT A2000 CARD - TAKES UP ONLY 1/2 SLOT
- EXCLUSIVE IVS CHECKLIST SOFTWARE MAKES CONFIGURATION A BREEZE
- EXCLUSIVE IVS HYPERDISK PORT CONNECTS TO IVS *WARPSPEED* DMA CONTROLLER CARD TO SPEED DISK I/O UP TO 8X!
- SIMPLE, USER CUSTOMIZABLE DRIVER INCLUDES SUPPORT OF OVER 15 SCSI DRIVES
- COMPREHENSIVE DISK UTILITIES INCLUDING DISK PARTITIONING

LIST PRICE: $189.95!
ASK YOUR DEALER TO CUT YOU IN ON THE BEST DEAL AROUND...
TRUMPCARD™
INTERACTIVE VIDEO SYSTEMS
15201 SANTA GERTRUDES AVE. STE Y102; LA MIRADA, CA. 90638 PHONE: (714) 994-4443

Starglider II

by Jeffery Scott Hall

Starglider II, from Rainbird Software, puts you in the cockpit of a futuristic fighter jet. Before the game starts, you must enter a password from the manual. Once the proper word is entered, the title screen is displayed and a stereo sound track plays.



The view from the Starglider II's cockpit.

The Pilot's Mission

It is your job to protect the planet of Novenia from invading Ergon forces. Not only is this planet at stake, but many others just like it are in danger. You must search out and destroy a vast space station that is being constructed. And that is not going to be an easy task. So strap yourself into the pilot's seat, energize those phasers, and prepare for some non-stop combat in outer space.

Examining the Starglider

You are seated inside the Icarus, otherwise known as the Starglider, looking at the view screen. Just below the view screen is a row of instruments and indicators in full 3D. These change and fluctuate with your ship for a visual confirmation of current status. In the center of the view screen is the heads-up display, used for targeting weapons and locking in objects for either identification or tractor beaming.

The Icarus has heavy offensive weaponry, including the plasma lasers, time-warp cuboids, fire and flee missiles, bouncing bombs, and neutron bombs. It's defenses include a super strong energy shield surrounding all sides of the ship, and an audio analysis of any sounds occurring around you. To keep you up to date on the status of the planet and your ship, at any time you can request a status report, which will be displayed at the bottom of the control panel. Now, it is time to see how the Icarus holds up in battle.

Going into Battle

You have just received word from the Novenian Council that the Ergon army has planned a full-scale invasion. After putting on your space suit, you board and launch the Icarus. Sitting in your ship, you start to wonder what freedom for your planet would be like; you are the only hope for this accomplishment.

Almost caught up in your dream, you notice the radar showing a massive fleet of Ergon forces closing in. Moving closer, you call up the target identification computer for more details about the enemy fleet. You quickly learn that most of the enemy forces are fighter jets, so you kick in the stardrive for a quick fly and take out as many targets possible. You cycle through your weapons trying to decide which one is best for this fleet. Aw, the plasma lasers ought to do the job.

As the two targets get closer, you grip the control stick tenaciously. Now, when you finally have them within range, you lock-on with the lasers to bring them both crashing to the surface. Suddenly, you hear a sharp whine coming from somewhere nearby. You turn on the audio analysis and discover that the sounds are coming from a homing missile launched from a Ergon ground walker. Unfortunately, it is too late to take any evasive maneuvers. The missile hits your left wing, dropping you from stardrive to normal speed.

Finding the ground walker before it has another chance to fire is your only chance for survival. As you lower your altitude, you select a fire and flee missile. This is your only option since damage has already been done to your ship. You finally locate the ground walker just ahead, and quickly increase your speed to maximum thrust and fire the missile. As you glance back, a bright light encircles the area where the ground walker once stood. Whew, that was a close call! You must now find a repair center. You know your mission has just begun, and the lives of billions are resting in your hands!

Summary

Starglider II combines some nice extra features that allow you to view the Icarus from any angle, whether you are in flight or on the ground. Two other important features are the abilities to pause and save the game to disk. Before now, I found it difficult to get engrossed in a flight simulator-type game. However, Starglider II combines great arcade action, strategy, graphics, sound effects, and music to keep you coming back for yet another daring flight into deep outer space!

Hints and Tips

Whenever you encounter an enemy fleet, kick in the stardrive and activate the plasma lasers. As you fly past them, take out as many space ships as possible. After doing so, deactivate the stardrive and lower your altitude to minimum level, so you are skimming the planet's surface. Now, as you fly past them again, take out as many ground forces as possible. Repeat this process until you have taken out most of the Ergon fleet. In some cases, this strategy may not work; before you attempt it, use either the audio analysis or target identification. If more than ten targets are on radar, do not engage the stardrive. Use your own tactics for survival.

When using the tractor beam, make sure the object mass is not too great for the Icarus to handle. If it is, a message tells you so, and the tractor beam must be immediately disengaged. Trying to use the tractor beam on an object too large destroys the Icarus and ends your mission!

Make sure you check the status report frequently, so you are aware of the planet's condition and any other information vital to your mission.

•AC•

Starglider II
Price: $44.95
Rainbird Software
P.O. Box 2227
Menlo Park, CA 94026

HARD DRIVES FOR AMIGA

Complete Hard Drive Units for the A2000 and A1000/A500
A1000 /A500 units come complete with SCSI Host Contoller, case w/power supply and Hard Disk
A2000 units come complete with DMA SCSI Contoller , cable and Hard Disk

A2000
20 Meg $539.00
40 Meg $749.00
65 Meg $799.00

A1000 / 500
20 Meg $639.00
40 Meg $799.00
65 Meg $879.00

External Floppy

3.5" with Metal case, extra length cable, full pass-thru, low power consumption. Fully Compatible with all Amiga Computers.
Single $159.95
Dual $329.95

Internal Floppy

A2000 only $139.95

COMPUTER MART

YOUR TEXAS SOURCE FOR AMIGA SOFTWARE AND HARDWARE
IMMEDIATE ACCESS TO OVER 500 PRODUCTS

CALL TOLL FREE **800-443-8236**

INFORMATION **409-560-2826**

COMPUTER MART 105 LYNN ST. NACOGDOCHES, TEXAS 75961

AMAZING PROGRAMMING

STRUCTURES

by Paul Castonguay

If you have taken more than a passing glance at C program listings for the Amiga, you have seen lines like these:

```
struct Window *wind;
struct NewWindow wnd;

wind = (struct Window *)OpenWindow(&wnd);
rp = wind->RPort;
```

Are you having trouble understanding what these lines mean? Have you been scared away from programming your Amiga in C because of them?

These lines deal with a concept called structures, and understanding them is essential if you want to do even simple things like opening windows and drawing graphics. In this article, I will explain structures to users already familiar with BASIC. To enhance the discussion, I will mention variables and arrays, but you should already understand them from your work in BASIC.

Most programming textbooks start this discussion by stating that structures are a data type. Data is information. Programs either use or generate data. A program that calculates your bank balance uses numbers as data. Your word processor uses letters and words as data. Because of the way computers work, data is stored differently depending upon whether it is numeric or alphabetic.

Also, if it is numeric, it depends upon whether the numbers are integers or decimals. The C language takes care of these details by providing you with variables of different data types. To store a letter, use a character type variable. Here "junk" is declared as a character variable and the letter Z is stored in it.

```
char junk;
junk = 'Z';
```

Here the number 23 is stored in the integer variable called "bunk."

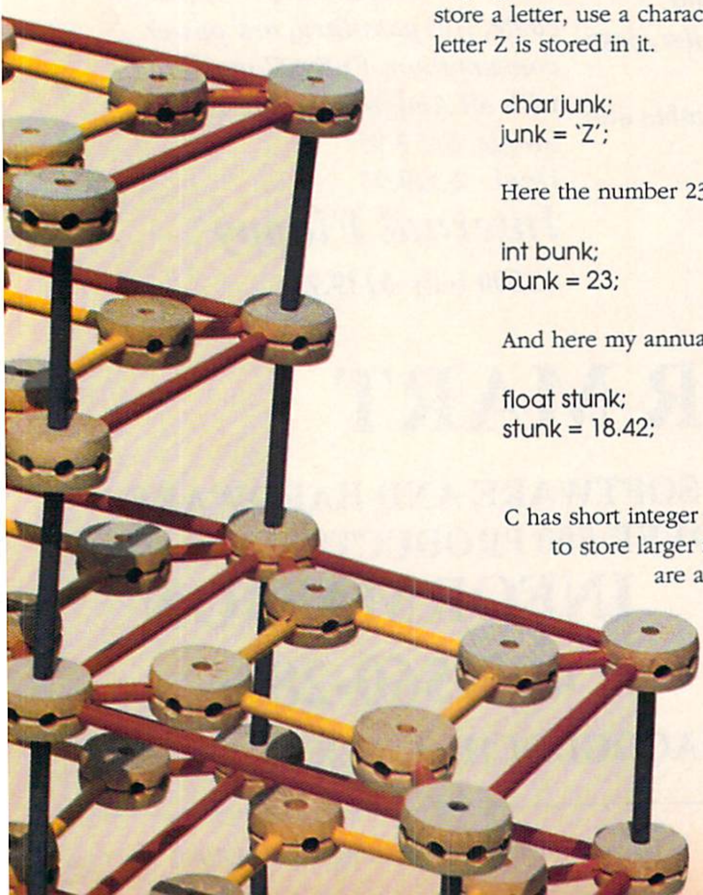
```
int bunk;
bunk = 23;
```

And here my annual salary is stored in the floating point variable "stunk."

```
float stunk;
stunk = 18.42;
```

C has short integer variables which let the computer run faster, and long integer variables to store larger numbers. Unsigned integer variables store only positive numbers. There are also single and double precision floating point variables:

```
float answer;
double e_base;
e_base = 2.7182818284590453;
answer = 2*3.1416*2.2E-06;
```



in C

A BASIC user's guide to programming C structures.

So... a data type is that attribute of a variable which determines the kind of data it can store. The data types mentioned so far are called fundamental, or elementary, data types. They represent the smallest units of each kind of data that can be stored. Each type is unique, and variables of each type will store exactly one piece of data.

BASIC also has data types:

STUFF$	is used to store words
JUNK	is used to store floating point numbers

You must have a good understanding of C's fundamental data types. I recommend *C Primer Plus*, by M. Waitz. Now I must press on or I will never get to structures.

Derived Data Types

These are data types which are made up of other data types. The simplest example is the array. An array can store a whole bunch of data of similar data types under one common name. Each unit (called an element) is individually identified by a special number called a subscript. Let me declare an array which can store ten integer values.

```
int score(10);
```

In it I will store my ten best bowling scores, one in each element of the array "score."

```
score(0)  = 86;
score(1)  = 92;
...
score(9)  = 103;
```

They form a group as they all have the common name "score." I can use that to my advantage if I print them.

```
for ( i=0; i<10; i++)
{
printf("My score in game %d was %d\n", i+1, score(i));
}
```

In C, the first element of the array is score[0], making the tenth element score[9]. Arrays can be any size or type you want, but most importantly, each element of a particular array must be of the same data type. In my example each element must be an integer. I cannot use the array "score" to store the name of my granddaughter's pet lizard. For that job I need a character type array:

```
char lizard(25);
strcpy(lizard, "Marmaduque");
```

Here I can store any lizard's name of 24 letters or less. C requires one extra element in the array to mark the end of the character string. Notice the use of C's string copy function, called strcpy(), to put a string into an array. You will have to study up on character strings in C if this appears difficult.

If I want to store many lizard names, I need a two dimensional array like this:

```
char lizard(5)(25);
```

Here I can store five lizard names.

```
strcpy(lizard(0), "Ninkin");
strcpy(lizard(1), "Stinkin");
strcpy(lizard(2), "Frog");
```

etc...

And you can print them out:

```
for(i=0; i<5; i++)
   printf("Lizard #%d is called %s", i+1, lizard(i));
```

In listing 1 you'll find an example that you can try out.

Compare C's arrays to BASIC's

```
DIM A$(25)
DIM X(100)
A$(3) = "My aunt Rose"
X(42) = 3.1416
```

If you don't understand arrays in C or BASIC, you're not ready for structures yet.

Structures

Structures are like arrays in that they are a derived data type, and that they allow many pieces of data to be stored under a common name. They are, however, more powerful than arrays in that they allow storing different data types. A single structure can store integers, characters, and floating point numbers, all under a single variable name. A structure can even store other derived data types like arrays and other structures. The Amiga's operating system often uses structures.

Each piece of data stored in a structure is called a member of that structure, and each member is referred to by a member name. This differs from arrays, which use subscript

Nothing but the best.

Now even better
with twice the
showroom space!

Still at...

396 Washington Street
Wellesley, MA 02181
(617) 237-6846

The Memory Location

Authorized Commodore
Amiga Dealer and
Commodore Service Center

numbers to identify each element. Let me define a structure which can store information about any of my girlfriends.

```
struct girlfriend
   {
      char first_name(25);
      char last_name(50);
      int age;
      char street_address(50);
      char town(25);
      int zip;
      float annual_income;
      long integer boyfriends;
   };
```

The above code is called a template for the girlfriend structure. The girlfriend structure is now a new data type which can store someone's first and last name, age, address, annual income, and number of boyfriends. The girlfriend structure is unique in that it will store only the kind of data that I designed it to store. If you were looking for a place to store your monthly bank statement, you couldn't use a girlfriend structure. Only one member of a girlfriend structure can store a floating point number. That's not enough for your bank statement.

One point of confusion in the above code may be the exact meaning of the word girlfriend. Girlfriend is not a variable name, it is a data type. I will not store data about any of my girlfriends in a variable called girlfriend. Instead, I will declare a variable to be of type girlfriend and then store data in that variable. Let me do this for three variables:

```
struct girlfriend best;
struct girlfriend worst;
struct girlfriend average;
```

Here I have declared three variables: best, worst, and average. They are each a separate variable, and each one can store all the information I want about a particular girlfriend. Each one is a girlfriend structure and has eight members with member names as defined in the template above. For example, best, average, and worst each have a member called first_name. We say that "best is a girlfriend structure." Similarly, "worst" and "average" are girlfriend structures. A girlfriend structure is a data type just like char, int, float, or any of the others. To use it, I have to declare a variable to be of that type, as I did above.

Referring back to my template for the girlfriend structure, notice that I, not the computer, decided exactly what kind of data a girlfriend structure would hold. To do that, I was required to make up names for each member of my structure and declare the appropriate data type of each. The member annual_income, for example, had to be a float type member because I wanted it to store a number with a decimal point. In each case I chose a descriptive member name, just as I would for any simple variable.

OK... so how do I go about actually storing data into a structure? I must refer to both the structure name (best, worst, or average) and the member name. There is a special notation which allows me to do that. First I write the name of the structure, the member operator, a period, and finally, the name of the member I am interested in. Let me store some data about my favorite girlfriend in the structure I declared as "best."

```
strcpy(best.first_name, "Gwendalyn");
strcpy(best.last_name, "Bonk");
best.annual_income = 145932.75;
best.age          = 31;
best.boyfriends    = 185;
```

The period is formally called the "structure member operator." I like to call this notation the "dot notation."

I can store the data in any order I want. Each piece of data is individually assigned to its respective member and each assignment has no effect on other members of the structure. I can assign data to the equivalent member of different structures like this:

```
strcpy(worst.last_name, "Kazookenberger");
strcpy(average.last_name, "Zonk");
```

Great! Now how do I access the data in these structures once I have put it there? I use the same structure member operator, the dot notation. Here a print statement shows what is stored in two members of the girlfriend structure called "best":

```
printf("My favorite girl is %s\n", best.first_name);
printf("She is %d years old\n", best.age);
```

Individual members of a structure can be used exactly like any simple elementary variable of the same data type. Here is a line from a program I wrote to update my girlfriends' salaries:

```
worst.annual_salary = worst.annual_salary - .06;
best.annual_salary = best.annual_salary + 17532.23;
```

Here floating point calculations are done to members of different structures just as if they were ordinary floating point variables.

One Last Time

A structure can store a bunch of data containing different data types under a common variable name, with each piece identified by its own special member name. The dot notation can refer to any member of the structure. BASIC doesn't offer structures as a data type.

One of the most powerful uses for structures is passing large amounts of data between functions. The Amiga people realized this when they designed the OpenWindow() function in Intuition. They knew programmers are fussy people who want almost every kind of window imaginable, different sized windows, with different locations on the screen, different gadgets, different colors, and so on.

How does the OpenWindow() function deal with all these requirements? Do we have to pass a multitude of arguments to the OpenWindow() function? No. Instead, we store all our window characteristics in a structure called NewWindow. The template for this structure has been conveniently placed in the header file <Intuition/Intuition.h>. Thank you, Amiga!

It's a smart idea to get a hard copy of this file and refer to it as you program your Amiga. In version 4.0 of Lattice C, this file is on disk #3. The file is long. I printed it using my word processor so I could add page numbers. The NewWindow template starts near the end of the file on page 17.

Notice the member names. LeftEdge, TopEdge, Width, and Height are so descriptive you could probably use them without any special instruction. But others, like IDCMPFlags and *FirstGadget, look rather intimidating. How do you use them? Don't worry about that just yet. At this point it is only important to know that a structure called NewWindow has been defined for you, that it has descriptive member names, and that you will be storing some data in those members using the dot member operator notation.

First we declare a variable to be of type NewWindow. Remember that NewWindow is not a variable name; it is a data type. To use it, we have to declare a variable of that type. Look in program listing #2 at the end of this article. At the top of the main() function is the line:

```
struct NewWindow my_window_request;
```

I have defined "my_window_request" as a NewWindow structure. Since this structure stores all the details about the kind of window I want to open, I gave it an appropriately descriptive name. You can call it anything you like. After all, it's your variable.

Now we assign each member the properties we want our window to have. Look further down in the listing. You'll see things like:

```
my_window_request.LeftEdge=50;
my_window_request.TopEdge=25;
```

This is the now familiar dot member notation we saw earlier. You can probably guess that these lines have something to do with where on the screen the window will appear. Other assignment statements like:

```
my_window_request.Flags = SIMPLE_REFRESH | ACTIVATE;
```

will leave you with that empty confused feeling. Be patient. Before we can learn the exact meaning of every member of each structure in the Amiga operating system, we must make absolutely sure that we understand what structures are and how to use them. For now, just accept that I have stored some data into the various members of the NewWindow structure which I called "my_window_request." Don't worry if you don't yet understand what the data means. Trust me, all the data is legitimate.

Notice, however, that some members, like my_window_request. Width, store numeric data, while others, like my_window_request.Title, store alphabetic data. The power behind structures is the ability to store different data types under one name, my_window_request.

Sometimes you will see a NewWindow structure in a program listing, initialized using a shorter notation which does not explicitly mention the member names. Do not use that notation. It is used by insecure software writers who believe their only chance for success in life is to write code so badly that no one, including themselves, will ever understand it. Use the member names for the same reason you use descriptive names for any variable; so your source code is more understandable for yourself and others.

OK, here's the part you've been waiting for. How do I pass all that data to the OpenWindow() function? Surprise! I do it with only one argument. I pass OpenWindow() the address of the NewWindow structure, which I called

THE HOTTEST AMIGA STORE IN THE COUNTRY.

Amazing Computer Systems is *hot.* Our sizzling selection of Amiga products has become the talk of the town. From more than 1000 software titles in stock to the hottest selections of hardware, accessories and books. All at red-hot prices. So remember. When you're hot, you're hot. And when you're not, you're not shopping Amazing Computer Systems.

Amazing Computer Systems, Inc.
Frankford Village Shopping Center
3030 N. Josey Lane #144 (2 doors South of Skaggs) Carrollton, Texas 75007 (214) 394-8383
Open Mon.-Sat. 10am-6pm Thurs. 10am-8pm

Amazing Computer Systems is not affiliated in any way with PiM Publications, Inc. or Amazing Computing.

"my_window_request." You see, the OpenWindow() function is very smart and already knows about NewWindow structures. All it needs to read and use all the data stored in any NewWindow structure is the structure's memory location. Effectively, all that data gets passed to the OpenWindow() function in one swoop. Here is the required notation:

```
OpenWindow(&my_window_request)
```

The & is the address operator in C. Prefixing any variable with the & operator passes the address of that variable, not the actual data it contains. Here, the address of the structure "my_window_request" was just passed to the function OpenWindow(). Knowing that address, the OpenWindow() function is able to read and use all the data contained in each of its members. The system responds by displaying a window with all the characteristics I asked for.

Two points worth repeating

Rather than pass all that data to the OpenWindow() function one member at a time, I simply told the OpenWindow() function where to find a structure which contained all the data. And I couldn't have done it without a structure, because only structures can store different data types under one name.

Pointer notation

Just when you thought you understood everything about structures, I ruined everything by introducing a new concept. Is this really necessary?

C uses pointers a lot. It's easier to pass data by referring to its address than by referring to the data itself. This was obvious when we called the OpenWindow() function and passed to it the address of our NewWindow structure "my_window_request." We did that by using the & operator. So why do we need pointers? The & operator works only with structures which have a variable name, and many times we need to refer to the address of a structure which has no variable name to refer to. This happens when the operating system passes information back to us in the form of an address to a structure.

When we executed the OpenWindow(&my_window_request) function, it went off and created a window according to the specifications we stored in the NewWindow structure "my_window_request." But suppose that now, after the window has been created, you wish to refer to some property of that window. Maybe you are writing a scaling routine and want to know the exact size of the window. You could refer to the original Width and Height members of the NewWindow structure used to call the OpenWindow() function in the first place.

But that may not work! Suppose the user resized your window after it was opened. He/she could certainly do that if you provided a re-sizing gadget. My example program at the end of this article does not allow that, but it could easily be changed. The Height and Width members of my_window_request represent the window's size when opened, not necessarily its size at some later time. To make my point even clearer, suppose you needed to refer to some other property of the window that is not even described by a member of the NewWindow structure. You will have to do that if you want to draw simple graphics in your window.

How are these things done on the Amiga?

In addition to making a window on the screen, the OpenWindow() function, creates a Window type structure. Look in <intuition/intuition.h>, on page 14, and you will see that a Window structure has many more members than the NewWindow structure you used earlier. You'll see members describing every possible attribute a window could have—its size, position on the screen, default colors, and so on. In addition to knowing all the member names for the NewWindow structure mentioned earlier, the OpenWindow() function is smart enough to know them all for the Window structure as well.

The OpenWindow() function also stores, in each one, information about the window it has just opened. Most importantly, the data stored in the Window structure is constantly updated by the operating system whenever the characteristics of your window are changed, either by the user or by your program. It is from this structure that you get the latest information on window size, window location, and many other features.

Where is this Window structure?

When you use the OpenWindow() function, it returns a number which is the address of the Window structure it created for you. It passes you the same kind of information you passed to it when you used the address operator.

But now, how do you use it? Up until now you have referred to data stored in a structure only by specifying the name of the structure, the structure member operator, and the member name you wanted. Unfortunately, the OpenWindow() function doesn't pass the name of the Window structure it created for you. (Remember, the word Window is not a variable name, it is a data type!) In fact, the Window structure the OpenWindow() function created doesn't even have a name! It has only an address. Hmm... You can't use the dot member operator notation, can you?

Sounds like a job for a pointer!

A pointer to a structure does two things. First, it stores the structure's address. Second, and more important to you now, it knows all the member names of the structure it points to and allows you to refer to them by using a special notation.

Let's talk about first things first. Look in my listing at the end of this article. At the top you will see the declaration:

```
struct Window *my_window_received;
```

I have declared a pointer to a Window structure. The * symbol is the pointer operator. This variable, my_window_received, will store the address of the Window structure which will be created when the OpenWindow() function is called. Here is how this is done:

```
my_window_received =
OpenWindow(&my_window_request);
```

This line not only creates your window, and a Window structure containing all the properties of the window just opened; it also stores the address of the Window structure in the pointer called "my_window_received." This is really no different than any assignment statement of a value returned from any function. Here are some more familiar examples:

```
x = sqrt(25.0);
char = getchar();
my_window_received =
OpenWindow(&my_window_request);
```

In each case, the value each function returns is assigned to the variable. The value returned by the OpenWindow() function, which happens to be the address of the Window structure, is assigned to the pointer variable "my_window_received."

Now, second things second! Each pointer variable knows all about the member names of the structure it was declared to point to. Pointers receive that knowledge when you declare them. You see, "my_window_received" was declared to point to not just any structure, but to a Window structure.

In addition to this, pointers use a special notation which allows you to refer to any member of a structure by name. I like to call this notation "little arrows." Let's use it to refer to something simple, like the size of a window. Look at your printout of <Intuition/Intuition.h> and notice the member names Width and Height in the Window template. Suppose I wanted to know what is stored in those members at any time. Here is the required notation:

```
my_window_received->Width
my_window_received->Height
```

The arrow is formed by typing a hyphen, followed immediately by the "greater than" symbol.

When you use this notation, you are telling your Amiga to "look in the structure pointed to by 'my_window_received' and tell me what is stored in the member called 'Width'." This notation can be used in assignment statements, just like the other dot notation you learned earlier.

```
wind_w = my_window_received->Width;
wind_h = my_window_received->Height;
```

Also, these members can be used just like any variable of similar type. Suppose you wanted to calculate the midpoint of your window for scaling purposes. Note that <intuition/intuition.h> declares both Width and Height to be of type SHORT. You might do something like this:

```
SHORT mid_x, mid_y;
mid_x = my_window_received->Width/2;
mid_y = my_window_received->Height/2;
```

Now you may use the values stored in mid_x and mid_y to plot a point at the exact center of your window.

Try not to feel overwhelmed by the long list of members in the NewWindow and Window structures. I know it all looks very complicated at first, but you don't need to understand the purpose of every member. Many members represent features of the operating system that you may never use; give these a simple default value like NULL.

Still, I know you are anxious to find out about them, so very soon I will talk about the meaning of one, the RastPort member of the Window structure. But right now it is important that you understand what a structure is and how to refer to data stored in it. Only when you feel comfortable with that can you begin to learn the exact meaning of any member of the many structures defined in the Amiga operating system.

Another **NEW** Product
To Make The **FASTEST** Development
Environment Even **FASTER!**

The **F-BASIC**™ Source Level DeBugger (SLDB)

The Only DeBugger Quick Enough For The **F-BASIC**™ System

Finally, a full featured DeBugger that doesn't require a working knowledge of assembly code. Use the SLDB to quickly debug your F-Basic programs without all those PRINT statements. Check these features of SLDB:

- Debug F-Basic programs at the SOURCE level.
- Fully windowed Intuition interface—windows for Source Code, 68000 Registers, Memory Dumps, Program Variables, etc.
- Set break points and single step trace by either source line or by Assembly Code Line.
- Display all variables, arrays, or RECORDS by name.
- Full reverse-assembler included.
- A comprehensive User's Manual with full documentation of all SLDB supported features and windows.

The **F-BASIC**™ Language System

An Enhanced Compiled BASIC Language
You've **READ** About It—Now **EXPERIENCE** It!

☐ Beginning AMIGA™ programmers choose F-Basic due to its simplicity and ease of use. If you know BASIC, you can use F-Basic immediately.

☐ Programmers of all levels choose F-Basic because its hundreds of built-in features provide an AMIGA language environment that coincides with your experience. F-Basic can't be outgrown, due to features like:

- Extensive Control Structures
- LOCAL & GLOBAL Variables
- Ultra Fast Floating Point
- RECORD Structures & Pointers
- INCLUDE & APPEND Separate Files
- PATTERN Matching Support
- Recursive SUBROUTINES & FUNCTIONS
- 32, 16, and 8 Bit INTEGERS
- Powerful String Facilities
- Direct ROM Kernel Access
- Bitwise Operations
- Access to 68000 Registers
- Easy high level access to AMIGA Screens, Windows, Menus, Sound, Speech, Graphics and Events.

☐ The F-Basic™ Language System includes:

- Comprehensive User's Manual—Full Documentation of All Supported Features Of The F-Basic System
- Sample Programs Disk—"Hands On" Tutorial
- FastCom™ single pass compiler, which quickly compiles F-Basic programs into assembly code. FastCom is written in F-Basic... That's your guarantee of the speed and efficiency of F-Basic.

The **F-BASIC**™ Language System—Only $79.95
F-BASIC™ System and Complete SLDB—Only $129.95

SEND CHECK OR MONEY ORDER TO:

DELPHI NOETIC SYSTEMS, INC.

Post Office Box 7722
Rapid City, South Dakota 57709-7722
Credit Card or C.O.D. Call (605) 348-0791

F-BASIC and FastCom are registered trademarks of DNS, Inc.
AMIGA is a registered trademark of Commodore/AMIGA Inc.

TxEd *PLUS*™

The Text Editor for the Amiga,™
Plus a whole lot more.

BlitzDisk Disk cache, speeds up floppy and hard disk reads up to 2000%.

FastFonts Speeds up text display.

FunKeys Hotkey window manipulator.

ARP Latest versions of the AmigaDOS™ Replacement Programs.

AREXX Demo version of the AREXX, the macro processor used by TxEd Plus, that is changing the way people think about computing.

TxEd

Complete package:
$79.95

MC + Visa
Mass Res. add 5%

Microsmiths, Inc
PO Box 561, Cambridge MA 02140
(617) 354-1224 BIX: cheath CIS: 74216,2117
Amiga and AmigaDOS are trademarks of Commodore-Amiga, Inc

Recapitulation

Two notations are used to refer to the members of structures. The dot notation uses a structure's name, the dot operator, and the member name. The pointer notation uses the name of a pointer to the structure, a little arrow, and the member name. Use the dot notation when you know the name of the structure. These are usually structures that you declare yourself in your program, like the NewWindow structure you called "my_window_request."

Use the pointer notation when you know the name of the structure's pointer. These are usually structures the operating system has declared for you, like the Window structure, whose address is stored in the pointer "my_window_received."

The Rastport

The first structure member you should learn on the Amiga is the RastPort pointer. The RastPort pointer is a member of every Window structure. It points to another structure which, incidentally, is also created when the system opens a new window for you. (A fine machine, that Amiga!)

Remember how I stated much earlier that structures could store any data type, even other structures? Here you have a Window structure which contains, as one of its members, a pointer to a structure called RastPort. Don't be concerned about this added complexity. The only thing you need to know about the RastPort structure is how to get its address as required by the graphic functions you want to use. You will do this with the pointer notation, the "little arrows." Suppose your program opens a window whose pointer is called "my_window_received." You probably used the OpenWindow() function like this:

```
        struct Window *my_window_received;
        struct NewWindow my_window_request;
```

Then, after loading my_window_request with data,

```
        my_window_received =
OpenWindow(&my_window_request);
```

The pointer "my_window_received" points to the Window structure the system created for you. To use the Move() and Draw() graphic functions in that window, specify the RastPort of the window in which you want to draw. You see the Amiga can easily draw graphics in many different windows which may be open on the screen at the same time.

It makes sense, then, that you, the programmer, must tell the system which window you want the graphics to be sent to by passing a RastPort address to each graphics function used. By specifying a RastPort address, you are telling the operating system which window you want the graphics to go to. The RastPort address for any window is stored for you by the operating system in the RastPort pointer, which is a member of every Window structure and has the name RPort. Here's how you send graphics to the window pointed to by "my_window_received":

```
        Move(my_window_received->RPort, 25, 10);
        Draw(my_window_received->RPort, 300, 90);
```

This will draw a line in your window. Many programmers find it easier to use a shorter variable name instead of "my_window_received->RPort":

```
        struct RastPort *rp;       /*    declare a pointer    */
        rp = draw-window->RPort;  /*  store RastPort address  */
```

Then you can use the more convenient notation:

```
        Move(rp, 25, 10);
        Draw(rp, 300, 90);
```

Let's go over this because it's important. Look again in <intuition/intuition.h> and find the declaration of the RastPort pointer. It's at the top of page 15 in my printout. It says:

```
        struct RastPort *RPort;
```

The name of the pointer is RPort and it points to a structure of type RastPort. RPort is the name of the pointer, RastPort is the data type that it points to. The expression:

```
        my_window_received->RPort
```

tells your Amiga to look in the Window structure my_window_received and see what is stored in RPort. That member has been defined by the template in <intuition/intuition.h> to be a pointer to a RastPort structure, and thanks to the designers of the Amiga, the number stored there is the address of the RastPort structure for your window. It is used as an argument for any graphics functions you want to use. It tells those functions in which window to draw their graphics.

Incidentally, every structure must have a template, and the RastPort is no exception. You can find the RastPort template in the header file <graphics/rastport.h>. It is automatically added to your program when you include <intuition/intuition.h>. As you develop your skills on the Amiga, you will want to look at the RastPort template.

One last point. You may have noticed that I have been careful to use upper case letters when required in the names of the NewWindow, Window, and RastPort structures. Remember that the C language is case sensitive, and these structures will not work if you do not write them correctly. Rastport is incorrect; it's RastPort.

Example Program

Listing #2 is a short program which opens a window on the Workbench screen and draws a triangle in it. Notice how the data for the window is first stored into a NewWindow structure by using the dot notation, member by member. Notice the Move() and Draw() function calls. They use a variable called rp to refer to the RastPort pointer. This tells the system which window you want the triangle drawn in. If you had opened two windows, and you are certainly free to do so, you could draw first in one window, and then in the other, by specifying the RastPort pointer for each window in turn.

Finally, look at the call to the OpenWindow() function. What's that (struct Window *) doing in the line which calls the OpenWindow() function?

The (struct Window *) is called a cast statement, and is there only because your compiler prefers to see it there. You see, the value the OpenWindow() function returns is an address—just a long integer. In your assignment statement, you store that number in a pointer to a Window structure. Your compiler sees this and wonders if you know what you are doing. It looks on the left side of the equation and sees a pointer to a Window structure. It looks on the right side of the equation and sees a long integer number. It recognizes that these data types don't match and sends a warning. Version 4.0 of Lattice reports, "Pointers do not point to the same object."

There is no problem, however, because you do know what you are doing and it is perfectly legal to store the address returned by the OpenWindow() function in the pointer "my_window_received." Actually, it's nice that your compiler recognizes the above data type mismatch because many times when you write programs, you will indeed not know what you are doing, and such a warning will alert you to a possible bug which would otherwise be difficult to find.

The cast statement (struct Window *) tells the compiler to treat the number returned by the OpenWindow() function, not as a simple long integer number, but as an address of a Window structure. The compiler says okay, assumes now that you know what you are doing, and continues without issuing any warning.

The cast statement (struct Window *) is not really necessary. You may write your program without it if you choose. I must admit, the line is easier to understand without the cast statement. If you leave it out, your compiler will issue a warning for that line of the program. Simply ignore the warning. If you put the cast statement in, you will not get the warning message.

Conclusion

I have discussed structures in the C language with an example of how they are used on the Amiga. I have explained how to store data into a NewWindow structure and how to read

Multi-Prefs

Time saving utilities for EVERY Amiga owner!

- Create multiple preference settings.
- Quickly and easily restore them as needed.
- Edit, Sort, Delete and Undelete any of them.
- Fast, Fun and Easy to use, from WorkBench or CLI.
- Runs on ALL Amigas . . . 500, 1000 and 2000 (V1.2 required).
- Multi-tasking.
- Not copy protected.
- Plus . . . 3 other time saving utilities are included.

Available NOW!!!

Just send $29.95 plus $2.00 S&H to:

SOFT·LINK inc. P.O. Box 304, Coventry, RI 02816

Dealer inquiries invited.
Amiga is a registered trademark of Commodore-Amiga, Inc.

STUFF-IT!

Design your own menus and use them with your favorite Amiga software!

- Each menu is like having thousands of user definable, descriptively labeled, function keys
- Quickly and easily create all the menus you like
- Modify any of your menus, as often as needed
- Each menu can restore other Stuff-It menus
- Also included are a few menus we've created to get you started (CLI-Menus, ED-Menus, etc.)
- Runs on ALL Amigas...From WorkBench or CLI (V1.2 or higher required)
- No hardware requirements • NOT copy protected

Stuff-It is a MUST for ALL Amiga owners, beginner to expert!

Just send $39.95 plus $2.00 shipping & handling to:

SOFT·LINK inc.

P.O. Box 304 Coventry, RI 02816
Amiga is a registered trademark of Commodore-Amiga, Inc.

data from a Window structure. What I have not done was explain, one by one, the exact meaning of each member of these structures. Throughout the article, I have asked you to not worry about this, emphasizing the more important goal of learning about structures. That done, now you probably want to know more about the meaning of each member of these and many of the other structures designed into the Amiga's operating system. However, I am not going to explain those to you.

Feel cheated? Don't. If you didn't understand structures when you started reading this article, and if you were able to follow everything I have presented, you have made a giant step in your understanding of the Amiga. You have conquered what I believe is the most difficult obstacle to programming the Amiga in C. You have learned what structures are. So, now what do you do? I know that opening simple windows will not hold your interest very long. Armed with an understanding of structures, member operator notation, and pointer notation, open your *AMIGA Intuition Reference Manual* by Addison-Wesley and see if you now understand what it is talking about.

Take the NewWindow structure, for instance. An explanation of each member starts on page 65. The first four are pretty easy. Specifying different values there will open windows of different sizes. Specifying illegal values will crash your machine! Fun, huh? DetailPen and BlockPen can be a pen number or a minus one. That's easy enough. But look at all that stuff for Flags! How are you going to understand all that? You won't, at least not right away.

What you must learn, however, is that all those names, WINDOWSIZING, WINDOWCLOSE, SIMPLE_REFRESH, and so on, are actually definitions in the header file <intuition/intuition.h>. Each capitalized argument is a macro which has been conveniently defined for you and which you are free to use by specifying it by name. (Thank you again, Amiga!)If you want to use more than one, and you usually will, separate each with the OR symbol, which in C is the character "|". (The | key is just left of the backspace key.) Use my program as a model to do the same in your own programs.

If you get stuck trying to understand any of these members, or find the explanation in the Intuition manual incomplete, try another reference. I suggest *Inside the Amiga* by John Berry, *Inside Amiga Graphics* by Sheldon Leeman, *Programmer's Guide to the Amiga*, by Robert Peck, and *Amiga Programmer's Handbook* by Eugene Mortimore.

LISTING 1: ARRAY EXAMPLE

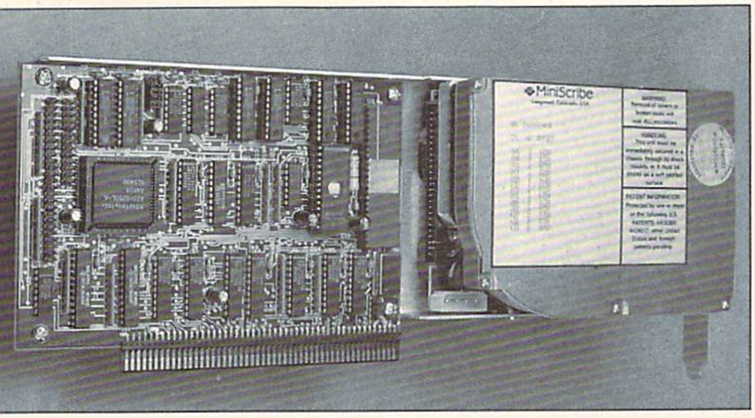
```
void main()
{

      int i;
      static turtle[25];
      static char lizard[5][25];

      strcpy(turtle, “Zakarious”);

      strcpy(lizard[1],”Blinkin”);
      strcpy(lizard[2],”Frog”);
      strcpy(lizard[3],”Clementine”);
      strcpy(lizard[4],”Matilda”);
      strcpy(lizard[0],”Stinkin”);

      for(i=0; i<5; i++)
      printf(“Lizard #%d is called %s\n”, i+1, lizard[i]);

      printf(“\nMy turtle’s name is %s\n”, turtle);

}
```

LISTING 2: OPEN A WINDOW ON THE WORKBENCH

```
#include <intuition/intuition.h>

struct IntuitionBase *IntuitionBase;
struct GfxBase *GfxBase;


VOID main()
{
    struct Window *my_window_received;
    struct NewWindow my_window_request;
    struct RastPort *rp;
    int i;

    VOID open_libraries();

    open_libraries();

    my_window_request.LeftEdge=50;
    my_window_request.TopEdge=25;
    my_window_request.Width=450;
    my_window_request.Height=125;
    my_window_request.DetailPen=-1;
    my_window_request.BlockPen=-1;
    my_window_request.Title=”THIS IS MY WINDOW”;
    my_window_request.Flags=SIMPLE_REFRESH | ACTIVATE ;
    my_window_request.IDCMPFlags=NULL;
    my_window_request.Type=WBENCHSCREEN;
    my_window_request.FirstGadget=NULL;
    my_window_request.CheckMark=NULL;
    my_window_request.Screen=NULL;
    my_window_request.BitMap=NULL;

    my_window_received = (struct Window *)
                         OpenWindow(&my_window_request);
    if (my_window_received == NULL)
    {
       printf(“I can’t open window !!\n”);
       exit(FALSE);
    }

/*  use variable rp so store the address of the RastPort  */

    rp = my_window_received->RPort;

    /*  Draw a black orange triangle  */

    SetAPen(rp,2);
    Move(rp, 25, 25);
    Draw(rp, 350, 45);
    Draw(rp, 155, 95);
    Draw(rp, 25,25);

    /*  Display an orange message  */

    SetAPen(rp,3);
    Move(rp,50,110);
    Text(rp,”Please wait... I’ll be gone soon”,32);


    for (i=0; i<1000000; i++);

    CloseWindow(my_window_received);
    CloseLibrary(GfxBase);
    CloseLibrary(IntuitionBase);
}

VOID open_libraries()
{
    IntuitionBase = (struct IntuitionBase *)
         OpenLibrary(“intuition.library”,LIBRARY_VERSION);

    if(IntuitionBase==NULL)
    {
         printf(“I can’t open intuition library !!\n”);
         exit(FALSE);
```

HardFrame/2000

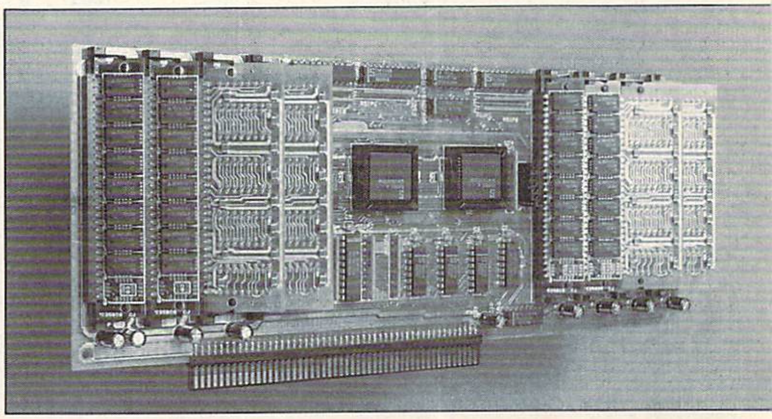
The Super-speed, DMA, SCSI Hard Disk Interface with 1.3 Autobooting



How fast is fast? **HardFrame/2000** transfers data at Amiga bus speeds! It's actually faster than the hard disk mechanism itself! And even more important in the Amiga's multitasking environment, **HardFrame/2000** has extremely efficient DMA circuitry to get on and off the bus in almost no time at all: 280ns to get on; 200ns to get off. **HardFrame/2000** autoboots under AmigaDOS™ 1.3 and is fully compatible with the new Fast File System. The core of any DMA SCSI interface is its SCSI protocol chip and DMA chip. MicroBotics has chosen the new, high performance Adaptec AIC-6250 SCSI chip, capable of up to 5 megabytes per second raw transfer speed, and the Signetics 68430 DMA chip running at 12.5 megahertz. Then we added additional FIFO buffering and enabled 16-bit wide data transfers for maximum throughput. The sophisticated design of **HardFrame/2000** provides for automatic SCSI arbitration, selection and reselection. The hardware supports either synchronous or asynchronous data transfer. **HardFrame/2000** can function as either the SCSI bus initiator or the target and can reside in a multiple master environment. Physically, **HardFrame/2000** is optimally flexible: the compact, half-size card comes attached to a full length, plated aluminum frame. The frame has mounting holes positioned to accept standard, 3.5" SCSI hard disk units such as those manufactured by MiniScribe, Seagate, Rodime, and others (hard disk mechanisms must be supplied by the user or his dealer as a separate purchase item). Alternatively, you can cable-connect to a SCSI drive mounted in your Amiga's disk bay or in an external chassis. As many as seven hard disks may be connected to a single HardFrame. There is no size limit on each disk. **HardFrame/2000** includes a 50-pin SCSI cable and header connectors for either 50-pin or 25-pin cable connection. Also included is a current tap to power frame-mounted drives directly from the slot itself. **HardFrame/2000** comes complete with driver, installation, and diagnostic software. Available September 1988. Suggested list price, $329 (hard disk not included).

The HardFrame/2000 photo shows the product with a MiniScribe 20 megabyte hard disk installed. Hard disks are not included in the purchase price of HardFrame. Note that if placed in the first slot, HardFrame uses only one slot.

8-UP! The Eight Megabyte Memory Card with Amiga™-specific DRAM Controller Logic



All the memory space you and your Amiga 2000 need –in a modern, highly integrated FastRAM expansion board. In **8-UP!**, MicroBotics went all the way to provide you with a truly Amiga-specific memory design to meet the special demands of the Amiga's high speed multitasking environment: The heart of any memory expansion is its *DRAM controller circuitry*. Rather than compromising with off-the-shelf parts, MicroBotics developed its own, custom controller design and built it into high-speed, *Programmable Macro Logic* chips (Signetics PLHS501). These new, super chips (each **8-UP!** uses two PML's) permit MicroBotics to employ *sparse refresh* technology to assure that your **8-UP!** is a truly zero wait-state/minimal-refresh-collision memory design. If you're putting eight megabytes in only one slot, that means that you probably have plans for your other A2000 slots. **8-UP!** gives you new freedom to do that planning since, unlike other ram peripherals, it is an extremely low-power memory card– a single, fully-loaded, 8-megabyte **8-UP!** draws an astoundingly efficient 800 milliamps! That's less than *two-fifths* of the power "budget" for a single slot! Low power draw also means that the card is cool-running for reliability and long life (not to mention a cooler Amiga!). **8-UP!** offers you maximum flexibility in memory configuration: it is organized into two separate *PIC's* (Amiga-speak for autoconfiguring peripherals). Each **8-UP!** PIC consists of four SIMM module sockets; these sockets accept either 256k-byte or 1 megabyte *SIMM's* (Single Inline Memory Modules). You can also purchase optional *PopSIMM* boards from MicroBotics; fill them with conventional RAM ; then use PopSIMM's to fill your **8-UP!** The card can run with as little as 512k of memory or as much as eight megs –with many intermediate configurations possible (particularly the six megabyte configuration, most desirable for use with a BridgeCard™). **8-UP!** is speedy, efficient, custom memory technology for your Amiga 2000 –and it's available now! **8-UP!** suggested list price is $199 (0k installed). Optional **PopSIMM's** are $49.95 per pair.

The 8-UP! photo shows the card half populated with conventional SIMM modules and half with MicroBotics PopSIMM's. PopSIMM's (without DRAM installed) are available as separate purchase items.

MicroBotics, Inc.

Great Products Since the Amiga Was Born!
811 Alpha Drive, Suite 335, Richardson, Texas 75081 (214) 437-5330
SOLD ONLY THROUGH YOUR AMIGA DEALER
Tell your dealer he can quick-order from MicroBotics directly – no minimum quantity –show him this ad!

"Amiga" is a registered trademark of Commodore-Amiga. "HardFrame/2000", "8-UP!", "PopSimm", are trade names of MicroBotics

CONQUER AmigaDOS
WITH MASTERING - CLI
FROM GRAPHIC Expressions

MASTERING-CLI is THE interactive DOS tutorial you've been waiting for. The best way to learn the CLI is to use it. Our software is designed to actively instruct as well as provide online help while you master the Amiga's Command Line Interface. A 2-in-1 package includes a full tutorial of AmigaDOS and a memory resident assistant, ready to help you with the most commonly used commands. Also covered are directories, batch files and much much more. Overcome the DOS barrier, ask your retailer for

MASTERING - CLI
A must for every new Amiga user

Or order direct, send check or money order for: $ 39.95 Postpaid in the Continental U.S. to:
GRAPHIC EXPRESSIONS P.O.Box 110028 Nutley NJ 07110
Amiga is a registered trademark of Commodore-Amiga Inc. NY/NJ residents add applicable sales tax.

```
void main()
{

        int i;
        static turtle[25];
        static char lizard[5][25];

        strcpy(turtle, “Zakarious”);

        strcpy(lizard[1],”Blinkin”);
        strcpy(lizard[2],”Frog”);
        strcpy(lizard[3],”Clementine”);
        strcpy(lizard[4],”Matilda”);
        strcpy(lizard[0],”Stinkin”);

        for(i=0; i<5; i++)
        printf(“Lizard #%d is called %s\n”, i+1, lizard[i]);

        printf(“\nMy turtle’s name is %s\n”, turtle);

}
```

```
#include <exec/types.h>
#include <intuition/intuition.h>

struct IntuitionBase *IntuitionBase;
struct GfxBase *GfxBase;


VOID main()
{
    struct Window *my_window_received;
    struct NewWindow my_window_request;
    struct RastPort *rp;
    int i;

    VOID open_libraries();

    open_libraries();

    my_window_request.LeftEdge=50;
    my_window_request.TopEdge=25;
    my_window_request.Width=450;
    my_window_request.Height=125;
    my_window_request.DetailPen=-1;
    my_window_request.BlockPen=-1;
    my_window_request.Title=”THIS IS MY WINDOW”;
    my_window_request.Flags=SIMPLE_REFRESH | ACTIVATE ;
    my_window_request.IDCMPFlags=NULL;
    my_window_request.Type=WBENCHSCREEN;
    my_window_request.FirstGadget=NULL;
    my_window_request.CheckMark=NULL;
    my_window_request.Screen=NULL;
    my_window_request.BitMap=NULL;

    my_window_received = (struct Window *)
        OpenWindow(&my_window_request);
    if (my_window_received == NULL)
    {
        printf(“I can’t open window !!\n”);
        exit(FALSE);
    }

/*  use variable rp so store the address of the RastPort  */

    rp = my_window_received->RPort;

    /*  Draw a black orange triangle  */

    SetAPen(rp,2);
    Move(rp, 25, 25);
    Draw(rp, 350, 45);
    Draw(rp, 155, 95);
    Draw(rp, 25,25);

    /*  Display an orange message  */

    SetAPen(rp,3);
    Move(rp,50,110);
    Text(rp,”Please wait... I’ll be gone soon”,32);


    for (i=0; i<1000000; i++);

    CloseWindow(my_window_received);
    CloseLibrary(GfxBase);
    CloseLibrary(IntuitionBase);
}

VOID open_libraries()
{
    IntuitionBase = (struct IntuitionBase *)
         OpenLibrary(“intuition.library”,LIBRARY_VERSION);

    if(IntuitionBase==NULL)
    {
         printf(“I can’t open intuition library !!\n”);
         exit(FALSE);
    }

    GfxBase = (struct GfxBase *)
         OpenLibrary(“graphics.library”, LIBRARY_VERSION);
    if (GfxBase == NULL)
    {
         printf(“I can’t open graphics library !!\n”);
         exit(FALSE);
    }
}
```

•AC•

Desktop Publishing with Professional Page

by Barney Schwartz

This month I will create a document using Gold Disk's Professional Page. Professional Page version 1.1 hit the streets in May of this year. As usual, Gold Disk has released two improvements to version 1.1 since May. First they offered a new metric file utility which greatly expands the postscript library in 1.1, allowing a much larger varity of fonts for PostScript users. Now they have introduced a new toolkit for dot-matrix users. This toolkit will build a metric file from any Amiga screen font file, giving you satisfactory output from Professional Page to your favorite dot-matrix printer. These new utilities and the 1.3 printer drivers which ship on version 1.1 help make Professional Page a desktop publisher for all users.

The new printer drivers increase graphics printing speed by 200 to 600 percent. They also allow multiple density printing and color correction, which lets you create good quality newsletters with your trusty nine-pin dot-matrix printer, or excellent quality flyers and articles from a laser output device.

Your readers will appreciate your efforts and you will no longer have to spend a great deal of money on printing services. You may, however, spend a lot of time waiting for your copies to roll off the printer. Even though 1.3 greatly enhances print speed, the time it takes you to get to a final product depends on your skill and knowledge of the publishing art. To this end, let us begin.

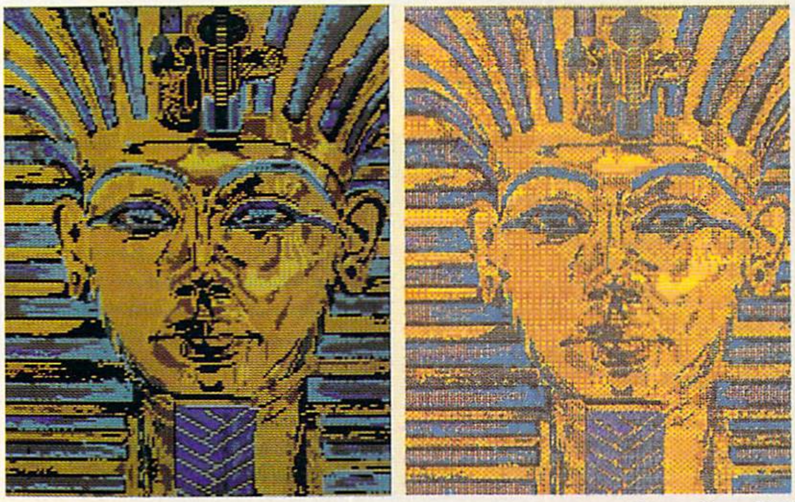


Figure One: TUT
Screen vs. Printout (Epson JX-80)

Look at figure 1. Those of you who have grown up with the Amiga will recognize it as TUT. It was done in Low-Res mode on Electronic Arts Deluxe Paint. For every piece of art which must be "pasted" into your article, two things must be considered:

1. Size of original versus allowable space.
2. Colors in original versus colors in final copy.

Those of you who are familiar with TUT realize that what you see here is not the same size or color as the original version. The screen colors don't match the output of an Epson JX-80 printer, and the original 5"x7" size was unacceptable for this column. As with almost every graphic, this piece required some manipulation prior to incorporation in the article.

Next, you must decide on your page layout. Will you have headlines? How many columns will you use? Will you require graphics? In which typeface and size will you set the body text? The Professional Page manual contains a wealth of information on all of these subjects. I find the discussion on column width particularly interesting.

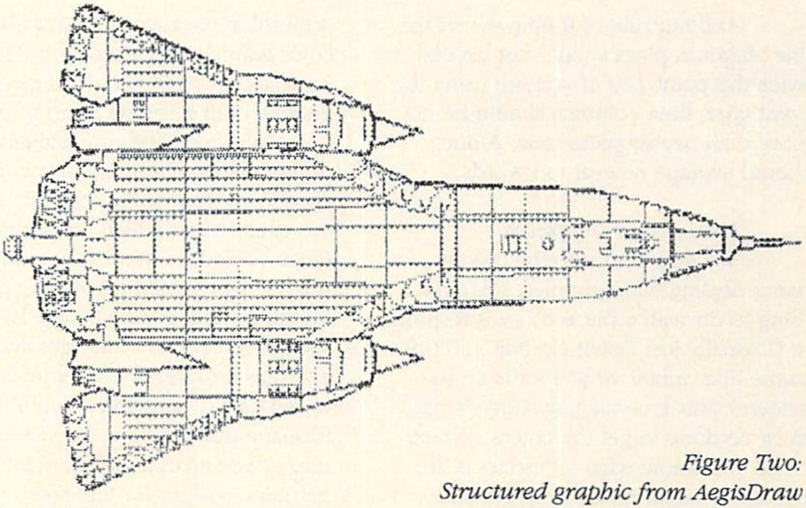


Figure Two:
Structured graphic from AegisDraw

"The easiest reading body copy (text of an article) should be set in at least ten point type in columns no smaller than one alphabet wide (a to z). An optimum length is one and a half alphabets. The alphabet is used to measure because it has balanced character widths. As a general rule, a column should not be wider than two alphabets.

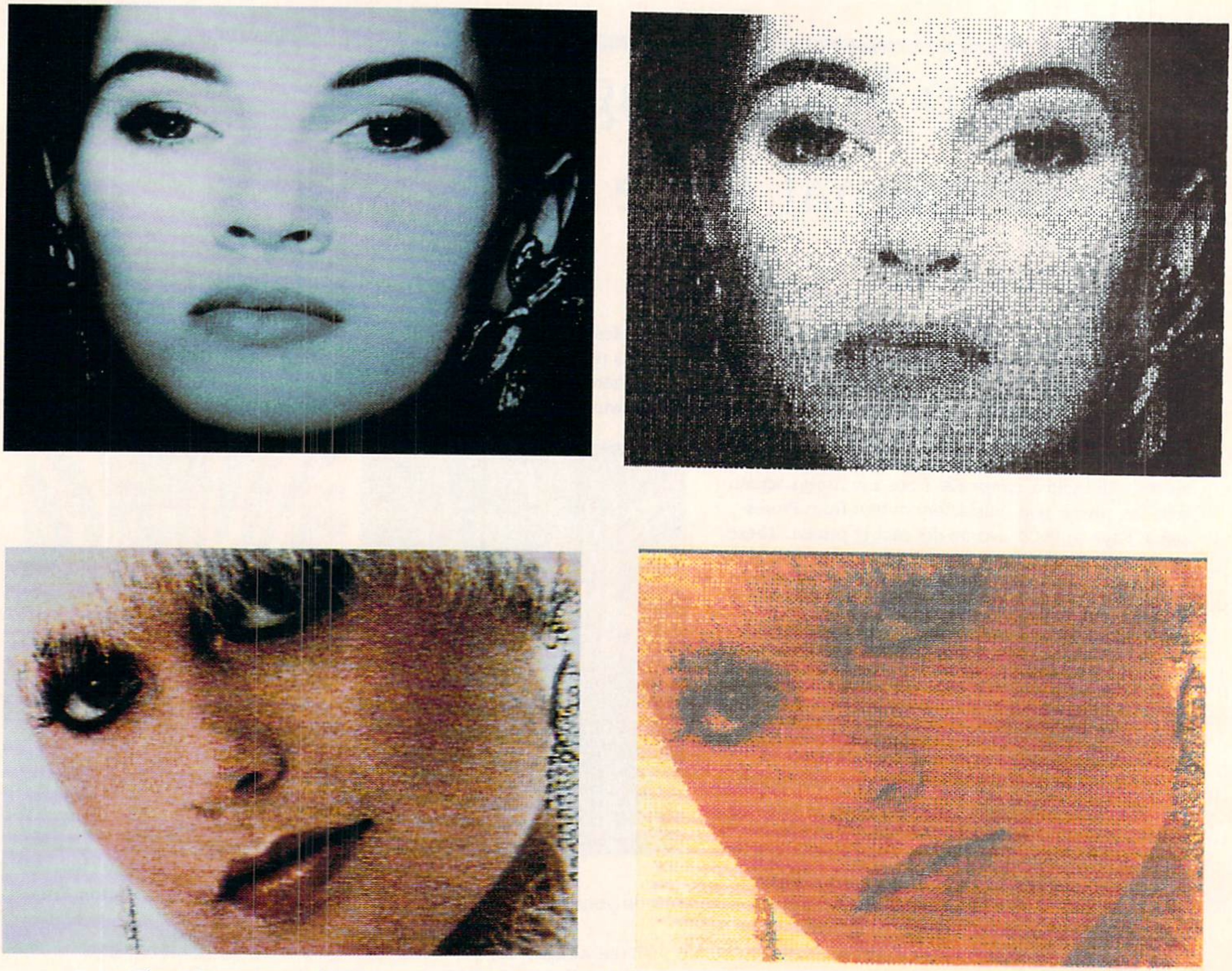


Figure Three: Comparison of IFF Images— As they appear on the Screen vs. Dot-Matrix (Epson JX-80) printout. (Top— B&W Bottom— Color)

Another rule of thumb is that the line length in picas should not exceed twice the point size. If you are using ten point type, then columns should be no more than twenty picas long. A line should average nine or ten words."

Bitmapped Graphics

We now return to the task at hand, dealing with graphics. The first thing to do with a piece of art is to print it. (I usually just install Grabbit and print to the line printer what ever is on the screen.) This immediately shows what must be done to get the colors correct. Once you know what adjustments are necessary, enter an art program and make the necessary changes before installing the graphic in Professional Page. This is necessary because Professional Page, like most other desktop publishing packages, doesn't display color graphics. As a matter of fact, with Professional Page you don't display your graphic until it is time to go to press.

This will save you valuable time during composition. Of course, if you choose not to display your pictures, you must remember which picture is where. Once you adjust the colors to fit your printer, you should place your graphic-file on an "issue" disk. Your "issue" disk should contain all of the graphics and text files required for the article you plan to publish. As a matter of habit, I usually format a disk, name it the article's name, make two separate directories (one for graphics another for text files) and begin planning. An "issue" disk is not required, but I have found it helps me stay organized, and saves disks swaps when I am finally ready to go to press.

Notice that I have said nothing about adjusting the size of the graphic. You may choose to put the graphic on the artboard or directly into your document. I prefer to put a blank box of the allocated size in the document and place the graphic on the artboard. This allows me to cut, crop, or resize the graphic to fit the allotted space. Once the graphic is cut or cropped, it will be sized to its proper dimensions. When you are satisfied the graphic is correct, transfer the finished piece to the document.

Graphic sizing can be accomplished from within Professional Page. Instructions for scaling a graphic are rather cryptic and obtaining the desired size may seem like a chore. But things are not that complicated. With the graphic box ACTIVE open the active box requester. Within this box you will see:

1. All margins inside the box. 2. The position of the box on your page. 3. Scaling factor in the X and Y directions (planes).

Use the scaling factor, which will be set to one initially. Adjust the factor until you have the correct size for your document. For example, suppose you start out with a graphic seven inches wide and five inches high. In your document, you want the graphic to be three and one half inches by two and one half inches. Set the X scale and the Y scale to point five. This will yield a graphic one-half the original size.

Structured Graphics

What are structured graphics? Look at figure 2. This is a structured graphic from Aegis Draw. It is not output as IFF bit-map, but as a series of vector lines. Professional Page inputs and scales these drawings much easier than IFF bit-mapped graphics. (This may be because the internal drawing utility is a structured graphics form.) There is usually no need to adjust colors for proper printer output, since the only colors are prime (i.e., RED, CYAN, MAGENTA etc.). The only draw-back to using this type of graphic is that your desktop publisher will not be able to read the stroke-font for lettering and measurements within the graphic. If this text is important, you may overlay boxes within the graphic. At any rate, you will probably appreciate the fine detail available from structured graphics.

Hold and Modify Graphics

Hold and modify (HAM) graphics are to me the most impressive type of graphic in the Amiga's large inventory of svelte tools. And they are so easy to use. You just open a box in the artboard, import IFF graphics (this time it's a digitized photo from Digi-Paint), size it to fit, crop and insert the graphic. If you feel color would be more effective for your graphic, and your graphic is black-and-white, you can shade it with one of the many image processors available. But that will be the subject of a future column. For now let me say that any of the following image processors would be well worth the investment:

1. Butcher, Eagle Tree Software.
2. Digi-Paint, NewTek.
3. Photon Paint, MicroIllusions.
4. PixMate, Progressive Peripherals

For an example of what you can do using digitized color graphics, see figure 3. This should convince you of the utility of color on a dot-matrix printer. If there is one thing a laser cannot do, it is color. True, lasers can sometimes do better text (but the best—typesetters do the best job). Lasers are also faster than most dot-matrix printers (once you get them printing). But they don't do color and they do cost big money.

Professional Page will accept any IFF graphic,including HAM. It will take color, black-and-white, or any image processed graphic. The only requirement is that the piece be IFF. There is one thing to consider when you work with such high resolution graphics: reducing size reduces detail. The more reduction the greater the loss of detail, the more detail you lose. This is not something to worry about, just don't expect a reduction to show every little detail.

Text & Word Processing

Text makes up most of any published document. Desktop publishing provides a means of incorporating text and graphics into a final page. But the desktop publishing package is usually not designed to perform word processing. This is not to say that you cannot word process with a desktop publishing package; you certainly can. However, you may find it much more convenient to write on a familiar word processor. After you've written your article, compose the pages in the desktop publishing package.

All true desktop publishers allow importation of text from many word processors. But some word processors are easier to transport than others. Perhaps the easiest text files to incorporate into formatted pages are pure ASCII files, files consisting of ASCII text characters only. Most word processors imbed format codes within their files. These imbedded codes facilitate proper output to printers from the particular wordprocessor. But these same imbedded codes wreak havoc on the composition artist.

It is possible to import formatted text from such popular wordprocessors as WordPerfect, Scribble!, and TextCraft, but even the best translation routines will not strip all of the format codes. This task is left largely to to desktop publisher. This is why I choose straight ASCII files to be imported into desktop publishing programs. When I say straight, I mean pure text strings: no paragraphs, no indents, no soft line ends,

the Analyst

For the "Well Adjusted" Amiga...

The Analyst integrates and builds upon the power of **WBExtras** (Workbench Extras) to provide a full complement of System Maintenance and Performance Optimization Utilities all from the Workbench. **Over 100 Windows** monitor every aspect of system operation, all with the ability to **Record, Print, and thus Compare Performance Attributes.** In addition, **The Analyst** provides...

- Easy assignment of **Function Key Sets** for quick access to any program with a single Key-Stroke
- Convenient DOS Path and Device Manipulation **including Device DisMount** essential for **Hard Disk Systems using Workbench 1.3.** This feature allows you to **"Boot-Up"** with the "New" **Fast Filing System, Console, and AUX Handlers** available as defaults
- Simple activation of any **Screen, Menu,** or **Interactive display** developed from your favorite Paint Program.
- Useful assignment of **Password Protection** for **Any Amiga Program**
- Enhanced Workbench Appearance with up to **16 Colors and Selectable Fonts**
- Simple Control of **Systems Lists and Task Priority** for added performance
- Easy Recovery of Discarded Files **Including the Icon**
- Complete Disk Verification including **File Integrity Validation**
- Time-saving execution of **System Maintenance Function** such as Disk-Verify and others while in **"Sleep Mode"** with time-defined **Screen Blanking**

Now Shipping
$89.95

WBEXTRAS

(Workbench Extras)
A Complete enhancement to the Workbench!

Now Shipping
$39.95

At **Luna C Software,** *we are dedicated to building* **Power Tools** *which allow* **"Your Creativity to become a Reality!"**

LUNA C SOFTWARE
P.O. Box 1388
Cañon City, CO 81212
(719) 275-5858

*Amiga and Workbench are Trademarks of Commodore-Amiga Inc.

68000 DISASSEMBLY

OTG Software

ANNOUNCING . . .

DSM VERSION 1.0d

DSM is a full-featured disassembler for the Amiga. Check out these features and you'll see why programmers agree, "DSM is the best disassembler currently available for the Amiga, bar none."

- DSM disassembles any Amiga program that does not use overlays.
- DSM produces assembler-ready output that is 100% compatible with the Amiga assembler, assem. Excellent results can also be obtained using the C.A.P.E. assembler from Inovatronics.
- DSM features true text detection.
- Improved disassembly is supported using the built-in expert system mode.

If your local Amiga dealer doesn't carry DSM, send check or money order to:

OTG Software
200 West 7th Street Suite 618
Fort Worth, TX 76102

TX residents add 7.25% sales tax. Price $67.50

no page breaks, no center codes, no justification codes. Just plain letters and numbers. The best way to accomplish this is to use your word processor's "save as text only" feature. Or you can use a text editor such as:

1. TextEd Plus, MicroSmiths.
2. MG1b, Fred Fish disks
3. MicroEMACS, Workbench 1.2
4. Lattice Screen Editor, Lattice.
5. Ed, Workbench text editor.
6. CygnusEd Professional, ASDG.

I use WordPerfect most of the time. This is because I'm used to it, not because it is better than the others. There are a multitude of really good word processors for the Amiga. (Each word processor is particularly well suited for a specific task: some are full desktop publishing, some straight word processsing, etc.) Anyway, after I write an article, and before I import to the desktop publisher, I strip all codes. Lately, though, I've been using TextEd Plus to do most of my work. A text editor eliminates the requirement to strip codes, saving valuable time.

TextEd Plus from MicroSmiths or CygnusEd Professional from ASDG, used with AREXX from William Hawes are two of the greatest text processing engines I've come across to date. I hope to discuss the capabilities of these programs in depth in a separate article. A bit of advice: if you haven't tried TextEd Plus or CygnusEd Professional, do so now. They are the only totally user configurable text engines. They can be whatever you want them to be. You can customize all aspects of their operation through the AREXX macro interface.

AREXX is another program that deserves a more attention. AREXX allows modular software to interact in a true multi-tasking environment. This allows for a module of one program to access modules inside another program, thus using available memory and time their capacity. As more modular software becomes available, and the industry expands to meet the challenge, the entire Amiga community will benifit.

Typestyles

What is type? What are the differences? What are the choices of output methods? What's so special about Gold Disk's Professional Page?

According to Webster, type is a rectangular piece of metal or wood with a raised letter in reverse on its upper end, which when inked and pressed against paper or other material leaves an impression of its face. So, "typeface" is the design of a particular letter in a particular font. It is safe to say that there are hundreds of thousands of typefaces. But there are only three basic typestyles: Serif, Sans Serif, and Ornate.

Serifed type is the typestyle most publishers will agree is the easiest to read. Therefore it is usually reserved for body copy, text which makes up the bulk of printed matter. Fine examples of serifed typefaces include Garamond, Baskerville, and Palatino. Serifed type has small "tails" or hairlines at the end of each vertical stroke in the letter. Theoretically, these serifs visually connect each letter, which provides for smoother, more comfortable reading.

Sans serif type does not have these hairlines on the vertical strokes. Each letter has the same line weight on all strokes. This gives each letter a blocky appearance which is less suitable for body copy but is ideal for attention grabbing headlines. Examples of popular sans serif typefaces are Helvetica and Avant Garde. The sans serif typeface doesn't form nice kern-pairs and is therefore hard to read .

Ornate type is any other typeface. Gothic, Old English and script are just a few ornate typefaces. The main function of these typefaces today is decorative only, and are sometimes used as oversized drop-letters to begin a paragraph. They are also used in place of italic or bold typefaces. When used in this manner they emphasize a word, meaning, or phrase.

Gold Disk's Professional Page with its new metric utility and screen font conversion tool lets you use any typeface in your library. These typefaces can be output to all standard dot-matrix printers and any postscript device. If your dot-matrix printer is not supported by preferences it will probably emulate one of the standards (Epson for instance). One word of caution on building a document based on an Amiga screen font: the document will not print correctly to a postscript device. But there is hope. If you are real clever, you will rename your font, its .font (dot.font) file and its metric file to a standard Professional Page font name. This will allow you to fool your system into thinking it can translate the file into postscript characters. For more information on this translation read the instructions whichaccompany the added metric files utility or call Gold Disk's tech support people.

With Gold Disk's continued dedication to product improvement I'm sure it is only a matter of time before we see virtual page compatability, rotated letters, true scalable fonts and other facilities. Professional Page is the closest thing in the Amiga's current arsenal to the current desktop publishing industry leaders. It's only a year old and growing fast. With continued attention, it is sure to become a system to be reckoned with. For more information on desktop publishing programs or answers to specific questions in the desktop publishing field write to:

Barney Schwartz
c/o Amazing Computing
P.O. BOX 869
Fall River, MA 02722-0869

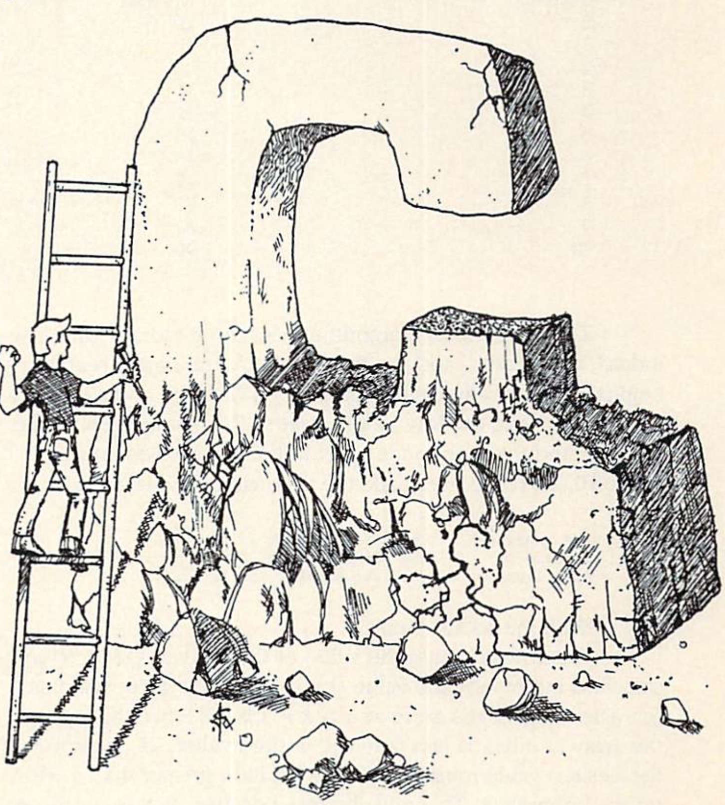
•AC•

On the Crafting of Programs

By David J. Hankins

Welcome back!

In last month's column, we began a discussion on optimizing programs for speed. The most logical starting point in optimizing is to determine what part of a program is responsible for its overall speed. We found this could be done using a software tool called a profiler. We ended last month's column by using a Manx C profiler (p1 and p2) to profile a simple C program.



This month we will apply the profiler to a real-world example—profiling the DSM disassembler Version 1.0c. The profile will help us as we attempt to increase DSM's speed. We'll also learn about two different types of searches, sequential and binary. And along the way we'll make an unexpected discovery about what mom tells us to do.

We'll begin by profiling DSM Version 1.0c. For those of you who are interested, the profiles discussed in this column were created by profiling the disassembly of the AmigaDOS program "diskcopy."

Following the instructions from last month's column, I produced the profile shown in Figure 1. Note, since the source code to DSM is not available to the general public (DSM is a commercial program), you won't be able to create this profile yourself. As shown in Figure 1, DSM spends 46.25% of its time in the function "check_reloc32". This function checks if 32 bit relocation was performed on a given address (the Amiga loader performs this relocation whenever a program is run).

Check_reloc32 checks to see if the address passed to it has been relocated by searching the array "relocates" (DSM stores the address of every longword which has been relocated in the array "relocates"). If the address is in relocates, relocation has occurred and check_reloc32 returns an index to the array. If the address is not in relocates, check_reloc32 returns -1 to signal that relocation did not occur at the given address.

Figure 2 shows the function check_reloc32. As you can see, this function searches the array relocates sequentially —the search begins at the first element in the array, and continues forward one element at a time until the address is found or until all elements have been checked.

I like to think of the sequential search as the "messy room search". If you have a messy room, you can't be certain that what you're looking for isn't in the room until you've searched everywhere. If you're lucky enough to find the object before you've looked everywhere, you can stop - you've found what you're looking for. So it is with the sequential search.

Fortunately for us, other kinds of searches exist. One of these searches is called the binary search. Binary searches are ordinarily much faster than sequential searches. To use a binary search on an array, the array must first be sorted. To illustrate the use of a binary search let's take, for example, the following array of numbers:

Unsorted Array of Numbers

Element	Number
0	15
1	7
2	12
3	99
4	16
5	11
6	3
7	5
8	19
9	21
10	80

Further, suppose we're searching the array for the number 21. If we were to use a sequential search on this array, it would take us 10 tries (0 though 9) before we found 21.

Before we perform our binary search, we sort the array from lowest to highest value. Doing so we get

Sorted Array of Number

Element	Number
0	3
1	5
2	7
3	11
4	12
5	15
6	16
7	19
8	21
9	80
10	99

The binary search algorithm uses three indices: the "low index", the "index", and the "high index." The binary search begins in the middle of the array; (0 + 10)/2 = 5, and so we set index to 5. Next, the low index is set to the first element in the array, 0, and the high index is set to the last element in the array, 10. To recap, we made the following initializations:

```
low index = 0     (the first element in the array)
index = 5         (the middle element in the array)
high index = 10 (the last element in the array)
```

With me so far? Good.

We now compare the value of the array at the location stored in index with the value (in this case 21) we're searching for. The value of the array at index = 5 is 15. Since the value of the array at index is less than the desired value, 21, we know the desired value must lie at some location greater than 5. How do we know this? This must be true because, as you recall, we sorted the array from lowest to highest.

Having determined that the desired value lies at some location greater than 5, we now change our low index value to 5 - the low index marks the lowest possible element at which we might find the value 21. We then choose a value for index which lies midway between the low and high indices. Thus index = (low index + high index)/2 = (5 + 10)/2. Rounding the value of index up results in index = 8. Let's review the changes we've made to index and low index:

```
low index set equal to previous value of index = 5
index set equal to (low index + high index)/2 = 8
high index was not changed and remains = 10
```

Again we check the value in the array at index. The value of the array at 8 is 19, still less than the desired value 21. Making the same changes to low index and index that we made before, we get:

```
low index set equal to previous value of index = 8
index set equal to (low index + high index)/2 = 9
high index was not changed and remains = 10
```

Once again we check the value of the array at index. We find that the value equals 21. We're done with our search!

Note that the binary search took us only 3 tries (we guessed elements 5, 8, and 9) to find our value, whereas it took us 10 tries (we guessed 0 through 9) with the sequential search.

To continue with the previous analogy, I like to think of the binary search as the "clean room search." In a clean room everything is stored in its proper spot. If you're looking for something, you can walk right up to the proper spot and find it. So it is with the binary search. We cleaned up the array by sorting it, and quickly found what we were look for.

Well, enough of this digression. Back to optimizing the DSM program.

In Figure 3, check_reloc32 has been rewritten to perform a binary search. As we discussed above, using a binary search instead of a sequential search should greatly improve the performance of check_reloc32. Note that before we can use the new binary search algorithm we must make sure that the array relocates is sorted. I made this change in a different part of the code, and used the qsort function from the Manx library to perform the sort.

How well did we do? Figure 4 shows the profile made of DSM after the sequential search was replaced by the binary search. As you can see, the check_reloc32 function went from taking 46.25% of DSM's execution time (see Figure 1) to only 2.58% (see Figure 4) of the execution time. Additionally, the non-optimized version of DSM took a total of 531,212.28 milliseconds to run (see the function _main in Figure 1) whereas the optimized version took only 304,059.91 milliseconds (see the function _main in Figure 4). This represents a 75% increase in program speed. Not bad for a day's work.

I guess all this talk about optimization has a moral to it when you really think about it. And the moral is this:

Remember when your mom used to tell you to clean your room? She was right! Next time she tells you to do something, pay attention—it may make you a better programmer.

That's all for this month. Next month I hope to take up the controversial topic of C-program format. Until then

```
void main(){ printf( "Goodbye world.\n" ) ; }
```

•AC•

***About the Author:** Mr. Hankins is the President of OTG Software, a small firm based in Fort Worth, Texas. It is rumored that Mr. Hankins can be found on-line on CompuServe (76515,1650), BIX (dhankins) or PeopleLink (OCS378) - that is he can be found on-line on those rare occasions when he is not otherwise engaged in making improvements to the DSM disassembler.*

LISTING ONE

```
              Self          Self + Children
# Calls   ms/call   %time   ms/call   %time   name
---  -----  --  -----  --  ----------

   2912      84.37 46.25        84.37 46.25   check_reloc32
  23014       2.10  9.10         5.82 25.22   format
 206108       0.21  8.10         0.40 15.60   aputc
 206108       0.13  5.08         0.14  5.42   putc
  12651       1.69  4.02         1.69  4.02   lookup_value
```

Don't limit your potential! Experience excellence!, a wordprocessor designed for your Amiga, with 250 available fonts, a Spell-As-You-Type 90,000+ word Dictionary, Grammatical/Style Checker, Thesaurus, Index and Table of Contents generator, Headers, Footers and Footnotes! Skate through PostScript output, True WYSIWYG, automatic Hyphenation, Math, beautiful resizable Color Graphics, flexible Mail Merge, Columns and an easy-to-use Macro-Language making complex actions more fun than a sleigh ride! The fastest wordprocessor for your Amiga is the only one you'll ever need! Have an excellence! holiday!

MSS
Micro-Systems Software
Committed to excellence since 1978

12798 Forest Hill Boulevard • West Palm Beach, Florida 33414 • 407-790-0770

See your local dealer or call our Sales Division 1-800-327-8724.
Amiga is a registered trademark of Commodore Business Machines • PostScript is a registered trademark of Adobe Systems, Inc.
excellence! is a registered trademark of Micro-Systems Software, Inc.

```
   3650        4.52  3.10         4.52  3.10  get_frag_seg_index
   1315       12.14  3.00        12.31  3.05  write
  68441        0.20  2.62         0.37  4.71  agetc
  87046        0.14  2.22         0.16  2.59  getc
   8307        1.24  1.94         4.25  6.65  fgets
    902       10.48  1.78        12.50  2.12  label_operands
    587       13.96  1.54        14.12  1.56  lseek

                            .
                            .
             (portion of profile omitted)
                            .
                            .

      1        4.30  0.00   531212.28 100.00  _main

                            .
                            .
             (portion of profile omitted)
                            .
                            .

      0        0.00  0.00         0.00  0.00  _abort
      0        0.00  0.00         0.00  0.00  dsm_err
      0        0.00  0.00         0.00  0.00  _exit
```

Figure 1: Profile of DSM Version 1.0c Before Optimization

```
/
*********************************************************************
 *
*
 *    function      :  check_reloc32
*
 *
*
 *    arguments     :  address = address at which relocation was done
*
 *
*
 *    return        :  index to array relocates[][]
*
 *
*
 *    discussion   :  check_reloc32 checks to see if relocation has
*
 *                     been performed on the longword residing at
*
 *                     address.  If so an index to the array
*
 *                     relocates, is returned, otherwise -1 is returned.
*
 *
*
********************************************************************/

check_reloc32( address )
unsigned long address ;
{
   int          i,                    /* counter
*/
                reloc32 ;             /* index to array relocates[][]
*/

   reloc32 = -1 ;
   for( i = 0 ; relocates[ i ] [ 2 ] ; i++ )
   {
      if( address == relocates[ i ] [ 2 ] )
         return( i ) ;
   }
   return( reloc32 ) ;

}
```

Figure 2: Sequential Search Algorithm

```
/
*********************************************************************
 *
*
 *    function      :  check_reloc32
*
 *
*
 *    arguments     :  address = address at which relocation was done
*
 *
*
 *    return        :  index to array relocates[][]
*
 *
*
 *    discussion   :  check_reloc32 checks to see if relocation has
*
 *                     been performed on the longword residing at
*
 *                     address.  If so an index to the array
*
 *                     relocates, is returned, otherwise -1 is returned.
*
 *
*
********************************************************************/

check_reloc32( address )
unsigned long address ;
{

   int             high_index,             /* upper bounds of search
*/
                   index,                  /* index to array
*/
                   low_index,              /* lower bounds of search
*/
                   printf() ;              /* C function printf
*/


/* 1.0 initialize indices */

   low_index = 0 ;
   high_index = num_relocates - 1 ;
   index = ( high_index + low_index ) / 2 ;

/* */


/* 2.0 search for value */

   while( address != relocates[ index ] [ 2 ] )
   {
      if( relocates[ index ] [ 2 ] > address )
         high_index = index ;
      else
         low_index = index ;
      index = ( high_index + low_index ) / 2 ;
      if( index == low_index )
      {
         if( relocates[ high_index ] [ 2 ] == address )
            index = high_index ;
         else if( relocates[ low_index ] [ 2 ] != address )
            return( -1 ) ;
      }
   }
   return( index ) ;

/* */

}
```

Figure 3: Binary Search Algorithm

```
               Self              Self + Children
# Calls   ms/call   %time    ms/call    %time   name
───  ────  ─  ────  ─  ─────────
  23014      2.12 16.08         6.10 46.17   format
 206108      0.21 14.16         0.43 29.19   aputc
 206108      0.13  8.92         0.14  9.55   putc
   1315     17.51  7.57        17.70  7.65   write
  12651      1.72  7.17         1.72  7.17   lookup_value
   3650      4.49  5.39         4.49  5.39   get_frag_seg_index
  68441      0.20  4.58         0.36  8.20   agetc
  87046      0.13  3.85         0.16  4.47   getc
   8307      1.23  3.37         4.23 11.56   fgets
    902     10.38  3.08        12.49  3.70   label_operands
    587     13.75  2.65        13.92  2.69   lseek
   2912      2.70  2.58         2.70  2.58   check_reloc32
      3   2532.20  2.50      4802.31  4.74   qsort

                          .
                          .
           (portion of profile omitted)
                          .
                          .

      1      3.84  0.00    304059.91 100.00  _main

                          .
                          .
           (portion of profile omitted)
                          .
                          .

      0      0.00  0.00         0.00  0.00   _abort
      0      0.00  0.00         0.00  0.00   dsm_err
      0      0.00  0.00         0.00  0.00   _exit
```

Figure 4: Profile of DSM Verion 1.0c After Optimization

•AC•

Desktop Video Part VI

Adding the Third Dimension

by Larry White

We've looked at many of the animation programs currently available, and so far, they all have one thing in common: two-dimensional movement. We can move an object up, down, or across the screen, but, except for a few special polyfonts in Deluxe Video and Aegis Titler, we can't rotate into the screen. Even though we've simulated a 3D effect by adding drop shadows and shading, the actual movement has been limited. Before we can enter the third dimension, we have to think in 3D.

Three Dimensional Thinking

What's so special about that third dimension? The complexity it adds to your animations may appear mind boggling at first. We can rotate an object along any axis, position the viewer at any point, and look at the scene from any direction. This means we must define what each side of our object(s) look like, where our 'lighting' is coming from, and even the object's surface (dull, shiny, etc.) so the computer can calculate the proper shading for each frame. The object itself may consist of many individually moving parts, such as head, body, arms, and legs; and we can move different objects in different directions. We can even move our viewer and zoom in and out while the objects are in motion! The calculations can get so complex that one meg of RAM may not do, and even the Amiga's 68000 microprocessor can often use help. If you plan to do a lot of 3D animation you'll want to pay special attention to the hardware enhancement box on page 53.)

Try the Real Thing

Since the phosphors in a picture tube lay on a single plane (although it may actually be slightly curved), the picture on a video or computer monitor can only really have two dimensions, height and width. Shading, perspective and other graphic techniques can create an illusion of depth in a drawing or a still graphic image. Yet, when we start moving the objects within that image, the shading and perspective must adjust with each frame or the realism will be quickly lost. This becomes even more obvious when multiple objects move, changing the spatial relationships between them.

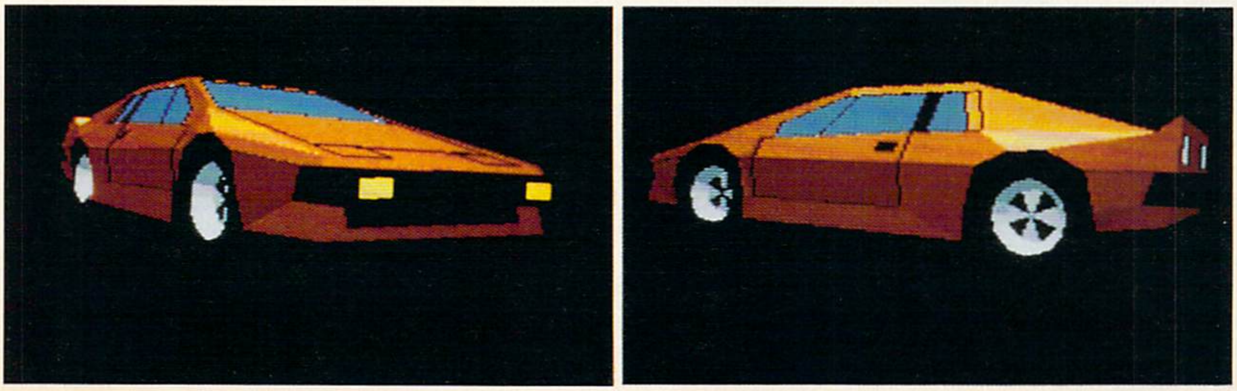
To better understand the 3D world, let's briefly step away from our Amigas, and turn on our video cameras. (A still camera with through the lens viewing and a zoom lens will also work.)

Place a model car in front of a plain background and place a lamp to the front side. Set the camera to a close-up so the model fills about 2/3 of the screen. Start the tape, pick up the car and rotate it in various ways. (Try to keep your hand from blocking the lens.)

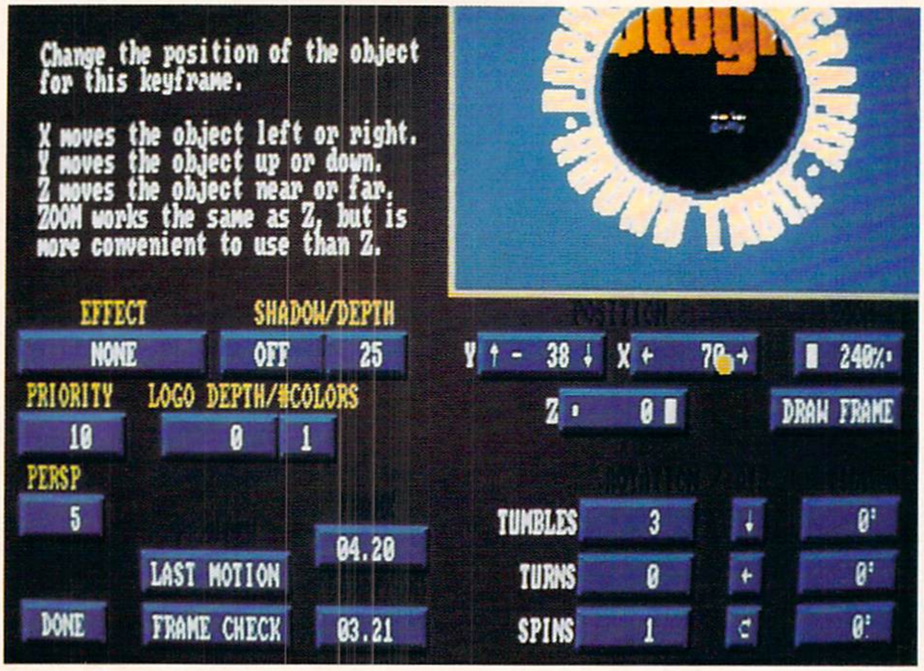
Playback the tape and examine how the parts closest to the lens seem elongated and emphasized. At some angles, the car's hood may seem distorted, and the parts closest to you will appear enlarged. Note how the shadows move as the car rotates, and how much detail there is on each side of the model.

Now place the model a long distance in front of a recognizable background. I placed mine on a railing and arranged the camera angle so that I could see a house about 1,000 feet away. Set the focus on manual and focus on the model. Now, zoom in and out on the car and notice how the background stays in the same proportions, although you will see less of it as you zoom in and more as you zoom out. (This is because as you zoom you change the angle of view, but not the perspective of the scene.)

Now switch to autofocus and zoom to a wide or middle setting. Keep the model centered in the viewfinder and

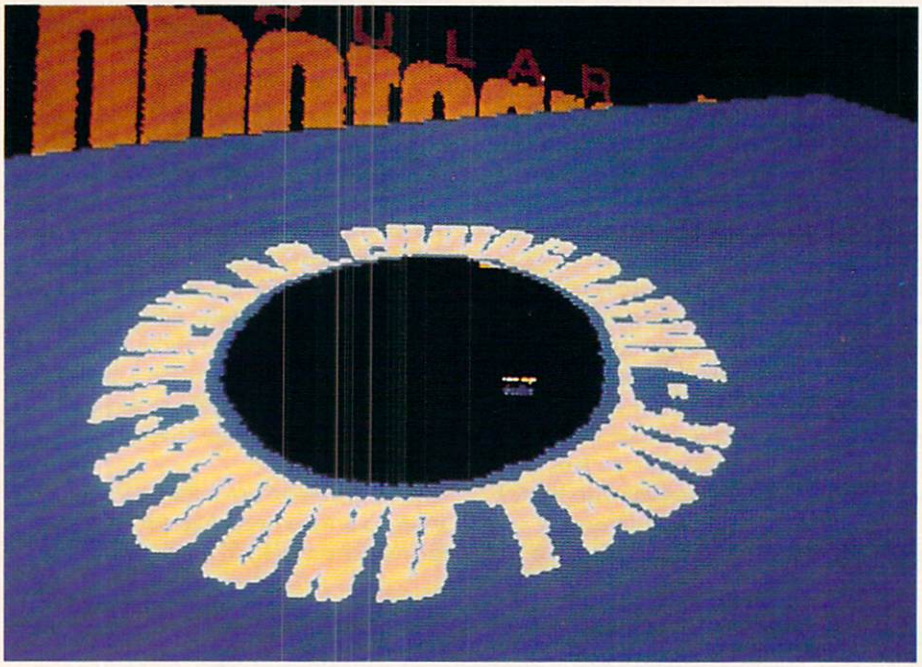


Two views of the infamous 3D Lotus— in VideoScape 3D



Change the position of the object for this keyframe.

X moves the object left or right.
Y moves the object up or down.
Z moves the object near or far.
ZOOM works the same as Z, but is more convenient to use than Z.



Video Effects 3D has a very convenient interface, but works only with overscanned Hi-Res IFF screens in 2,4, or 8 colors.
(Bottom: Snapshot of an animation sequence created with Video Effects 3D)

walk toward the car. (Unless you use a tripod on a wheels, your image might appear a bit shaky, but if you concentrate on holding the camera steady, the results should be sufficient for this exercise.) Notice how the background changes this time. Now back away from your model and watch again as the perspective changes. The closer the background is to the subject, the more these changes will be apparent. In film and video making this type of camera movement is called a dolly shot.

Now think of what would happen if the car were moving. To keep the car a constant size on the screen, the camera would need to follow the car at the same zoom position and distance. It would have to move parallel to the car, in the same direction and at the same speed. Film and video techies call this camera movement trucking.

As the subject moves, its relationship to the lighting changes, moving shadows and highlights, and complicating matters further. As if all this weren't enough, if you intend to place the animation over live video via genlock, you'll want to make sure that you match the lighting to avoid having objects with shadows on opposite sides. As we enter the world of 3D animation we'll need to keep these in mind to add realism to our computer generated videos.

The next time you watch television, pay special attention to the relationship between objects (people, cars, props, etc.) and the background during various scenes. Notice how they move relative to their background. Pay attention to camera motion and the sharpness (or lack thereof).

Entering the Third Dimension

Let's start by moving a 2-dimensional object around in a 3-dimensional world. This technique is extremely useful for producing effective titles.

You can create a title screen using almost any paint or titling program that saves your art as an IFF file. Before you start, check which resolutions and screen sizes your animation package supports. Also check for any special limitations on palette size or selection.

Two 3-D animation programs are Video Effects 3-D (InnoVision Technology) and Animation:Effects (Hash Enterprises). Animation:Effects can use any resolution IFF picture including overscan. You can even use HAM in low or medium resolution modes (although you can't mix resolutions at the same time). Personally, while I found this program very effective, I was not impressed by the user interface. However, I expect an updated version soon, which claims a vastly improved user interface. Video Effects 3D has a very convenient interface, but works only with overscanned Hi-Res IFF screens in 2,4, or 8 colors. Both programs can create dramatic animation effects, but the actual processing time for your final image can be rather long, overnight is recommended.

I needed a short title sequence for a videotape of a roundtable discussion sponsored by Popular Photography Magazine. I wanted to use the actual magazine logo and a special logo for the "roundtable series" and have the title sequence blend with the live video at the end. I decided to try Video Effects 3D.

Beginners

Have you ever seen a GURU MEDITATION ALERT or a TASK HELD REQUESTER telling you that a program has crashed?

If you had GOMF in your system, you would have been able to remove the program that caused the error, and you may not have had to reset your computer!

"GOMF performs flawlessly...[it] is not a luxury, it's a necessity.....perhaps the most useful utility yet introduced for the AMIGA." -Amiga World

Hypertek/Silicon Springs
205-2571 Shaughnessy Street
Port Coquitlam, B.C. CANADA
V3C 3G3
Dealers: call for special prices!

GOMF

the gurubuster !

NEW VERSION !!!
It's amazing! Now, you can actually save data AFTER a program has crashed! (Even after a total lockup) We call it the GOMF BUTTON.

GOMF BUTTON	$74.95
GOMF SOFTWARE	$39.95

order toll-free!
1-800-663-8526
Technical Support 604 524-1125

experts

- Restores your memory and system resources after a crash.
- Remove tasks, processes and devices
- Use from WB or CLI.
- Includes pages of technical reference to assist you with your programming.

"Anything that can stop this terrible [GURU] syndrome deserves a round of applause"
- Commodore Magazine

ATTENTION PROGRAMMERS!
Do you have a great idea for a product for the AMIGA? Give us a call for more info!

Other Offerings:
Deluxe MIDI Interface. Simply the best you can buy! Only $95
TTL monitor interface $99

Let's Make Some Titles

You can create the titles using any titling program (such as TV*Text, or Aegis VideoTitler) or using any paint program that can save the title screen as an IFF file. I chose to start with a high contrast, black & white copy of each logo, digitized them using DigiView. Just as I was getting started, a package arrived containing DigiView 3.0. In addition to supporting overscanned HAM images, DigiView has a special mode for digitizing just this type of "line art".

The program operates like version 2.0; but specifying this mode forces all colors either black or white; producing a very sharp Hi-res copy of the artwork. When using this mode with overscan, I had some difficulty, so I decided to keep to standard screen. I could change it to overscanned later by loading the logo as a brush on an overscanned paint screen.

To prepare the digitized artwork for animation, you can use an image processing program such as Pixmate or Butcher, and/or a paint program such as DigiPaint or Deluxe Paint II. Make sure to set the proper resolution and screen size for your animation program. Deluxe Paint II requires a 704 X 452 page size. This is not the standard Dpaint overscan, but can be entered on the "page size" menu by specifying width and height.

For multiple graphics to appear simultaneously, make sure to use the same palette. If you don't, objects on the screen will suddenly change color as new objects are introduced in the animation sequence. Although you must define the entire screen for Video Effects 3D, during the animation, you can crop into the graphic screen and treat any part of the screen as if it were an IFF brush. If you plan to genlock over live video you'll need to watch where you place color 0, since this area will be replaced by the live video.

3D Effects

Video Effects 3D requires at least 1 megabyte of RAM but 1.5 Meg is recommended. Depending on which Amiga model you have and the type and amount of expansion ram, you may need to change the stack and execute a SloMemLast command in your startup-sequence. The manual is very specific and easy to follow.

Before you run the program, format 2 disks. Label one "Script Disk" and the other "Play FIle Disk". One minor drawback of this program is that you can't format with the program running. Also, if the program needs the disks it will not prompt you. Since the processing can take hours, it might be some time before you discover your error and have to start over, a very frustrating situation.

The basic principle of this type of animation is simple. You define the first and last frame of your sequence, and a few specific frames in between. You then specify what happens to your object between these "keyframes"; and specify how long the sequence should run. Your Amiga will then calculate each frame of the sequence and record the frames as an IFF Anim file for playback.

When you run Video Effects 3D you'll get a slightly different type of Amiga-user interface. A screen of requester-type boxes and a round yellow mouse pointer will appear. There are no pull-down menus or Amiga Hotkeys, but if you position the pointer over one of the requesters a block of text will change to explain the current requester—no clicking required. Click on the requester and the program goes into action.

First, define a script (or specify an existing one to edit). Then select various IFF pictures as your objects. Each object will form a separate track which will contain the motion of that specific object. Several tracks can be running simultaneously allowing you to manipulate many objects in the same scene.

The 3D Universe

Before we can even begin to animate we need to take the subject and place it in a 3D universe. A standard monitor screen can be thought of as a two dimensional grid. Conventionally we call the screen width the X axis, and the height the Y axis. In many of our 2D animations we used X-Y coordinates to specify the screen location of our objects. We can specify the movement perpendicular to the screen as the Z axis.

We can specify the location of the object on the screen using only the X and Y coordinates. Changing the Z coordinate will move the object towards or away from the viewer.

Now we can turn the object in 3D space. If we specify 'tumble' the object moves in much the same way as a log rolling down a hill. You can see that the object is rotating on an imaginary pin placed along the X axis. If the pin were along the Y axis, the object would 'Turn' like the page of a book. Placing the 'Spin' is equivalent to turning a page upside down then rightside up. (Our imaginary pin would be pointing directly out of the screen). You can specify any number of tumbles, turns, or spins, and even have an object perform more than one at a time. For example, making an object tumble and spin as it approaches the viewer can be a very dramatic way to introduce a title.

By specifying the center of rotation for your object, you can get various effects of tumbles, turns, and spins. If you moved the center of rotation to the left edge of a block of text a turn effect would make the title move like a hinged door.

It's often difficult to picture exactly what an effect will look like. I often reach for a small block of wood and turn and tumble it before instructing the program to proceed. You can use a 1/4 screen wire-frame mode to preview an action between keyframes.

When you've programmed all your objects (tracks) you can place your playfile disk in the proper drive and begin rendering. Your Amiga will be tied up for quite some time, so be patient and prepare to do something else. When rendering is finished you can use the player program to view your work.

The resulting animation is usually amazing. Don't be surprised if the result is not exactly as you predicted. Even the most expensive animation systems can produce unexpected results (although usually with much faster turnaround time), and sometimes these results are even more spectacular than you expected.

Video Effects 3D has a few more tricks we haven't discussed which can enhance the 3D effect, like adding drop shadows to your animation. You can even use the program to extrude a two-dimensional logo into a three-dimensional one (as if it were carved from a block of wood) then use the 3D logo in your animation!

This brings us to the topic of the next installment— 3-dimensional objects and the methods for creating them in your computer (including ray-tracing programs such as Silver and Sculpt-3D).

•AC•

Suppliers:

Aegis Development
2210 Wilshire Blvd., #277
Santa Monica, CA. 90403

VideoTitler, VideoScape 3D

Brown-Wagh Publishing
16795 Lark Ave., Suite 210
Los Gatos, CA. 95030

TV*Text, Express Paint

Commodore Business
Machines
1200 Wilson Drive
West Chester, PA. 19380

Expansion Memory for Amiga

Eagle Tree Software
P.O. Box 164
Hopewell, VA 23860

Butcher

Electronic Arts
1820 Gateway Drive
San Mateo, CA 94404

Deluxe Video, Deluxe Paint II

Hash Enterprises
2800 East Evergreen
Vancouver, WA 98661

Animation:Effects

InnoVision Technology
P.O. Box 743
Hayward, CA. 94543

Video Effects 3D

Impulse Inc.
6860 Shingle Creek Pkwy. #110
Minneapolis, MN 55430

Silver, Turbo Silver

Mimetics Corp
16360 Stevens Canyon
Cupertino, CA 95014

Frame-Grab Board

NewTek
115 West Crane St.
Topeka, KS 66603

DigiView 3.0, DigiPaint, Video Toaster

Pacific Peripherals
P.O. Box 14575
Fremont, CA 94539

Subsystem,
Amiga Expansion Products

Progressive Peripheral & Software, Inc.
464 Kalamath Street
Denver, CO 80204

Pixmate, Frame Grabber

VIDEO SIDEBAR

Special Hardware?

The number of calculations in a 3D animation or ray trace can be astounding. On my A500 with 1 meg, a fairly simple object can take over 24 hours to be rendered at the highest resolution with anti-aliasing activated. I've added an expansion chassis from Pacific Peripherals which can accommodate 2 A2000 boards.

In one slot I've put Commodore's new 8 megabyte board (populated with 2 meg of fast RAM) which has helped speed things up a bit (I've already reserved the other slot for a video frame grab board - such as NewTek's Video Toaster or Mimetics' board (both expected soon)).

I've been told that the new 86020 and 86030 chips plus math coprocessors can cut the calculation time dramatically, often to just a few hours.

Not all software can utilize the coprocessing; if you choose this route make sure that the program versions you intend to use support the processor and coprocessors.

-Larry White

COMING SOON!

Look for the Amazing Product Guide: Winter '89 on dealer shelves in December!

A Complete Resource of Anything & Everything for the Amiga!

The Amazing Product Guide: Winter '89 will be a complete descriptive listing of *all* products available for the Amiga. If you have appreciated the theme-specific product guide installments in past issues of AC, just imagine bigger and better chunks of information. The *Amazing Product Guide: Winter '89* will feature longer, more detailed descriptions and even more products!

For only $4.95, you can have concise listings of the entire Amiga market at your fingertips! Just reach back to your bookcase, and you have all the information you need. No fluff, no extras. Just the facts necessary to make the smartest purchase for your needs.

Where? When? How?

The Guide will be available on newsstands in December for $4.95. All our dealers, from Europe to Australia, from Canada to the depths of the darkest forests, will receive limited supplies of the *Amazing Product Guide: Winter '89* along with their December shipment of AC. From there, it's up to you.

Don't Delay!

The *Amazing Product Guide: Winter '89* will be a must for all Amiga users (fire-breathing or not), but your Amazing Dealer will sellout quickly! Amazing Computing readers digging for the most useful information in the Amiga market will make sure the shelves are cleared in no time.

So please don't delay—this resource of anything & everything for the Amiga is a tool that no Amiga user should be without.

THE BUGS & UPGRADES COLUMN

BUG BYTES

by John Steiner

This month's list of bugs and upgrades begins with a mystery. A friend and I have the Amiga 2090 hard drive controller card, and we have had some trouble with overscan mode and hard disk access. We have also noticed a thread on People Link, regarding others having similar problems.

If you are a regular reader of Bug Bytes, you may recall that there have been a couple of upgrades to the software that comes with the 2090. The latest of these upgrades solves the problems my friend and I have been having. Problems occur in overscan mode, and they typically manifest themselves as irregular read/write errors on the hard disk. It seems that information on the drive can be damaged, as I have been forced to reformat the disk on more than one occasion when the problem occurred.

While reading the People Link message base the other day, I noticed several messages from people still having trouble with hard drive errors in overscan mode. In fact, several even had problems with high resolution 640x400 mode in some software packages. The commonality in the problem relates to SCSI based hard drive systems, and there still seems to be a problem with drive access for those using SCSI drives. The drives used by my friend and I are ST-506, the IBM-type hard disk.

I hope not too many 2090 owners are unable to use the Amiga in hi-res and/or overscan mode. If you have a 2090 and a SCSI drive, I am interested in whether or not you are having problems in high resolution or overscan modes.

Comments left on People Link usually say the problem occurs due to the behavior of the Amiga bus when the display is in hi-res, four bitplane mode. In this mode, chip memory is not available to the CPU or to a DMA device during the unblanked display period. If the CPU or a DMA device tries to access chip memory, it is forced to wait until the horizontal blanking interval—the only time chip memory is available. For each scan line of 63 microseconds, horizontal blanking lasts about 13 microseconds.

The 2090 disk controller is a DMA device, and therefore must deal with the above situation. The expansion software that comes with the 2090 evidently has been modified well enough to handle the ST-506 type drives, but from the above reports, it seems there still is a problem with SCSI-type units.

If you have a 2090 card and a SCSI drive in your A2000, please drop me a line via Amazing Computing, or at either of the nets listed below. Let me know whether or not you are having problems in high resolution or overscan modes. Please list the software packages you are having problems with; it seems that not all software works improperly with the SCSI hard drive/2090 combination. Also, several people with memory expansion boards were not having any problems *if* they used FastMemFirst as the first command in the startup sequence. There could be many other variations to this problem, so please list *all* peripherals you are using.

A few weeks ago, I purchased Deluxe Productions from Electronic Arts and immediately had a problem with hard drive access with the program. The program runs in high resolution overscan mode. A call to EA technical support brought some assistance—primarily the names of DProductions' developers at Associated Computer Services. When problems occurred, I didn't realize they might be related to the FastMemFirst problem, but now I am convinced of it. The technical support staff recommended that since I have expansion memory, I should put FastMemFirst in the startup sequence file *before* BindDrivers. This action cleared up the problem immediately.

On the subject of Deluxe Productions, I also noticed some minor problems in the documentation. One problem dragged me through several Guru visits. The manual's Appendix includes an excellent section for DPaint II users interested in making the most of using Deluxe Paint with Deluxe Productions.

There is one error, however. The manual states that you specify the page size with the Page Size command in the pictures menu. It further comments that you can select Standard (640x400), Full Page (640x480), and Full Video (672x444). Full Video is recommended for pictures created in Deluxe Productions. Actually, Full Video mode in Deluxe Productions is 704x480. A pair of boxes is available where you can type in the exact page size, if it is not listed.

I had not really noticed the difference between what the documentation said about page size, and the size that Deluxe Paint listed; I simply chose Full Video. The results: pictures created for Deluxe Productions only occasionally loaded or were accessible. The machine visited the guru, and other strange or unusual symptoms cropped up. When I discovered the discrepancy in screen size, I edited the pictures to the 672x444 format that Deluxe Productions recommends. The program worked flawlessly.

By the way, the manual also states that Deluxe Productions accepts a picture larger than 672x444, but will not display any larger image. Based upon my own experience, this statement is evidently not true. At least, Deluxe Productions does not reliably accept a Full Video image from Deluxe Paint.

The FlickerFixer board from MicroWay is an exceptional product. The board is a high resolution display adapter that provides a 640x400 non-interlaced (flickerless) high resolution mode. The hardware requires a VGA or multi-sync monitor, and provides a beautiful Amiga image in any resolution mode. The latest version of the flickerFixer is now compatible with the European PAL versions of the Amiga 2000.

MicroWay
Box 79
Kingston, MA 02364
(617) 934-2414

Version 1.4 of ACCESS!, a shareware communications program, has recently been released. A new feature is the ability to read and use script files. Occasionally, though, the program cannot find a script file. ACCESS! seems to have trouble getting the actual drive location from the window that comes up with the on-screen gadget. Using the pull down menu item to get a script shows that the program is looking on the wrong drive. Change the drive location to the one you want, and the gadget finds the proper drive. ACCESS! is on most public domain bulletin board systems.

GOMF from Hypertek/Silicon Springs has been upgraded to version 3.0. GOMF helps the Amiga recover from system Guru messages. The latest version of GOMF can be purchased with the GOMF Button, a hardware attachment installed under the Paula chip. A small wire with a pushbutton is brought outside the Amiga, so it can be accessed after a Guru message. When the system locks up, the button releases the errant task and unlocks the machine. The GOMF 3.0 Button with software retails for $69.95, and the GOMF 3.0 software-only is 34.95. Registered owners of previous versions can upgrade to the hardware version for $30.00, and the software-only upgrade is $10.00.

Hypertek/Silicon Springs
205-2571 Shaugnessey St.
Port Coquitlam B.C.
Canada V3C 3G3
(604) 942-4577

Lightning Logger from RTS software has been upgraded to version 3.0. The program now has faster, smaller code and can log to RAM:, RAD:, VDK:, or VD0:. If you are a registered user of the original, send your old disk and $5 for version 3.0 to:

RTS ELECTRONICS
P.O. Box 9015
Cocoa Fl.32922
(407) 631-9286

City Desk is now being upgraded to version 2.0. The latest version includes text flow around irregular graphics, auto hyphenation, and PostScript support. Gray scales up to 16 shades are available on the output, along with four shades on the screen display. Registered City Desk users can upgrade for $35.

MicroSearch, Inc.
9896 Southwest Freeway
Houston, TX 77074
(713) 988-2818

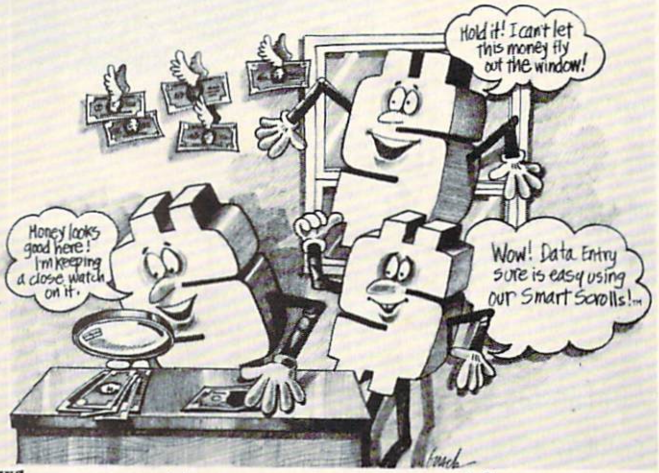
That's all for this month. If you have any bugs to report, or if you know of any upgrades to commercial software, you may notify me by writing to:

John Steiner c/o Amazing Computing
P.O. Box 869
Fall River, MA 02722-0869

...or leave EMail to
Publisher on People Link or
73075,1735 on CompuServe

•AC•

Meet a team of the friendliest financial organizers you'll ever run across.


Hold it! I can't let this money fly out the window!
Money looks good here! I'm keeping a close watch on it.
Wow! Data Entry sure is easy using our Smart Scrolls!™

When you want to manage your personal finances, Money Mentor goes a step beyond.

Plug Money Mentor into your Amiga and a virtual teamwork effort takes place in watching over every aspect of your personal finances.

The new "C" version of Money Mentor is the friendliest financial organizer obtainable today!

Now you can experience super-speed data entry, dazzling graphic output and an extremely friendly attitude!

Smart Scrolls for speed.

Money Mentor has a truly unique system called *Smart Scrolls*, that handles a diversity of otherwise tedious data entry functions and clips along saving you up to 70% of your typing time. It's a *smart* addition to Money Mentor, that's why we call it *Smart Scrolls*.

Money Mentor Features:

- Net Worth Statement
- 200 Budget Categories
- 30 Integrated Accounts such as Checking, Cash, Savings and Credit Cards
- Elaborate Search Routine allows editing of transactions according to your specific guidelines
- Automatic Check Printing
- Automatic Account Balancing
- Color Graphic Reports illustrating *actual vs. budgeted* amounts
- Over 50 Reports to choose from!

What they're saying about us!

"Money Mentor has to be the nicest look and feel of any money manager package for home use that I have ever seen." — Amiga Sentry

"Money Mentor is an excellent product" — Amazing Computing

Money Mentor is for everyone!

It does more than just keep your checkbook balanced. Money Mentor helps you manage your personal finances which is important to any family or individual. With Money Mentor, you can be looking better financially. Order Money Mentor today.

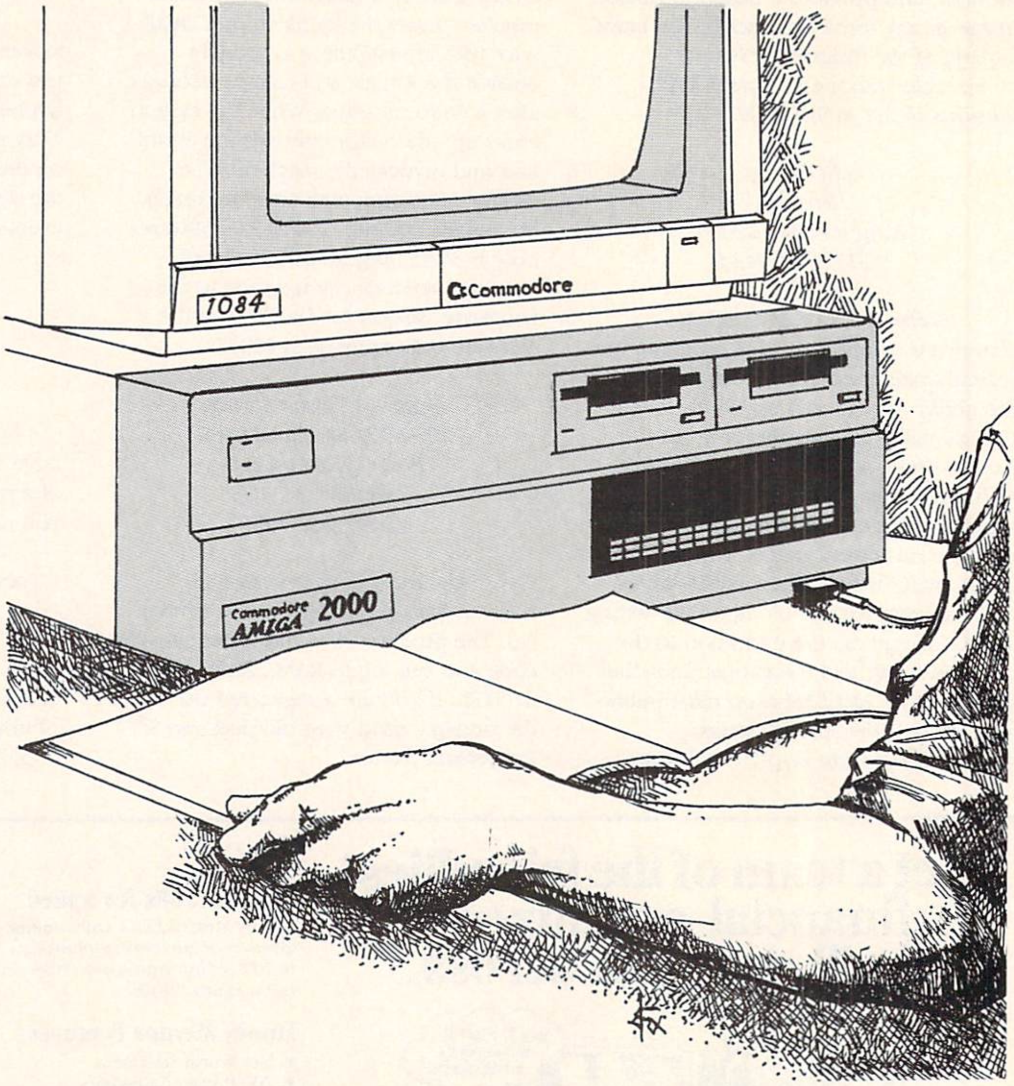
Money Mentor sells for only $95.95!

VISA MasterCard

SEDONA SOFTWARE

SEDONA SOFTWARE/11828 RANCHO BERNARDO RD., SUITE 128-20/SAN DIEGO, CA 92128/CALL (619) 451-0151

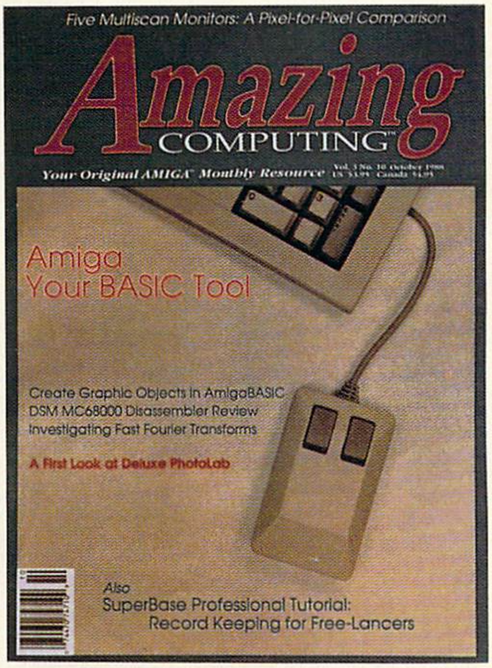
For the Amiga user who has everything...



What do you give the Amiga user who has everything? Well, there is one thing that *every* Amiga user desperately needs—solid, useful, practical information. And you know there's only one source for that kind of information—your Amiga monthly resource, Amazing Computing.

Each month, Amazing Computing connects creative, ambitious users to the power behind the Amiga. The Amiga has the potential to perform in endless computing arenas. Amazing Computing provides the insight, detail, and examples you need to make the Amiga fit your needs and your dreams.

An AC Gift Subscription!



Five Multiscan Monitors: A Pixel-for-Pixel Comparison

Amazing COMPUTING™

Your Original AMIGA™ Monthly Resource

Amiga
Your BASIC Tool

Create Graphic Objects in AmigaBASIC
DSM MC68000 Disassembler Review
Investigating Fast Fourier Transforms

A First Look at Deluxe PhotoLab

Also
SuperBase Professional Tutorial:
Record Keeping for Free-Lancers

Ho, Ho, Ho!

AC goes beyond magazine staples such as product reviews, trade show reports, and interviews to get down to the core of users' needs. Sure AC publishes unbiased, highly-regarded reviews and tasty industry whispers, but we really concentrate on hands-on applications. Step-by-step hardware projects, desktop video applications, down-and dirty-tutorials, and coverage of all levels of programming keep Amiga users reading AC each and every month. And for fun-lovers, AC also provides thorough coverage of those great Amiga games and other entertainment options.

Want to make an Amiga user happy year round? This year, give your Amiga enthusiast a priceless gift: solid, useful Amiga information. That rare commodity, information you can really put to work, can be found in only one source—your Amiga monthly resource, Amazing Computing.

2000

Amiga 2000 Hard Drive Roundup

by Sheldon Leemon

Last year's 20 megabyte hard drives cost over $1000, and ran only about 3-5 times as fast as a floppy disk. This year, you can purchase a 65 megabyte hard drive system for the A2000 which runs from 5 to 10 times faster than last year's models; and sells for under $1000. And soon, you'll be able to boot your computer directly from the hard drive, instead of from a Workbench disk.

The speed increases have been made possible by the Fast File System, a software change that was one of the main agendas of Workbench 1.3 (which we all fervently hope will be available by the time this is printed). We have already detailed the optimizations used to increase the speeds in our the hard drive roundup of AC V. 2.6. To use the FFS with a hard drive, you must copy the FastFileSystem file to the L: directory of your boot disk, change a few entries in the Devs:Mountlist file, and then reformat the drive.

Not only does the new file system increase speed many times, it also increases capacity slightly, since all 512 bytes of each data block are used for data, instead of the 488 bytes available under the old file system. This means an extra 50K of storage for each 10 megabytes of drive capacity. Although it's technically feasible to use the FFS on floppy disks, it's still quite cumbersome, and the speed gains are not anywhere as dramatic as those achieved by hard drives. Kickstart 1.4 should overcome both these limitations; and it will include the FFS as the default file system for both floppies and hard drives.

Kickstart 1.3 also offers autobooting. In fact, the only difference between Kickstart 1.2 and 1.3 is the autoboot code added to 1.3. Autoboot allows the computer to start up from devices such as hard drives, network cards, and even Commodore's new recoverable RAM disk, without having to have a floppy in drive df0:. This means that only hard drive owners who want to autoboot will benefit from getting the new 1.3 ROMs.

But just getting a new Kickstart ROM will not automatically cause your hard drive to autoboot. The hard drive interface card must include boot code in ROM. None of the controllers currently sold come with these boot ROMS. Some controllers have sockets on the board, just waiting for the new Kickstart ROMs to be distributed. Even some of the interface boards which don't have these ROM sockets may be retro-fitted to enable auto-booting. Since the FFS must still be read from a disk before a Fast File partition can be mounted, the boot partition must be formatted with the old file system. Therefore, most autobooting drives will have at least two partitions, a small boot partition and a large Fast File partition.

There are a number of reasons for today's lower prices. The Amiga 2000 made it possible to install a hard drive internally, thus eliminating the expense of an external case and power supply. The 2000's internal expansion capabilities have also made it easier for manufacturers to enter the Amiga hard drive market, since they need only produce an interface card and driver software. Increased competition between manufacturers has tended to move prices lower.

Finally, the market prices for hard drives, particularly the SCSI hard drives often used in the Amiga, have decreased while capacities have increased. As a result, Amiga owners will no longer have to pay twice as much for hard drives as owners of IBM compatible systems do. Amiga SCSI drive systems cost only about $150-$200 more than the IBM ST-506 drive systems, and mounting one on the Amiga costs about the same as mounting a fast SCSI drive in an IBM.

To DMA or not to DMA

The interfaces tested use two general types of data transfer schemes, DMA and processor-controlled I/O. In processor-controlled I/O, the 68000 microprocessor is used to read each byte of data from an I/O register, and then copy it to a memory buffer. Direct Memory Access (DMA) transfers, however, don't require the processor to fetch each byte. DMA devices can write directly to a memory buffer within the Amiga, without processor intervention. Normally, DMA devices and the 68000 alternate their control over the bus, so that neither interferes with the normal-speed operation of the other.

Theoretically, DMA devices can transfer data without slowing down any of the processor's normal tasks. This approach can lead to significant performance improvements in a multi-tasking machine like the Amiga, where time saved by one task can be used by another. In fact, it's just this kind of "co-processor" approach that make Amiga graphics acceptably fast within a multi-tasking context.

Unfortunately, a number of real-world conditions prevent the user from getting DMA data transfers "for free". For one thing, each transfer requires some processor overhead, as the driver tells the DMA chips how much information to move, and where to transfer it. After the data has been moved, the system must be informed that it has arrived. Therefore, a DMA interface demonstrates its greatest speed when moving large amounts of data. The overhead involved may make it less efficient to use DMA to move small amounts of data than plain old processor-controlled I/O.

Another potential problem with DMA is that there are only so many bus cycles to go around. Normally, the 68000 processor controls the bus half of the time, and the other half is available for

DMA devices. Unfortunately, the hard disk isn't the only device competing for the bus. The display processor, the blitter, the sprites, the audio channels, and the floppy disk drives all use up part of the available system bandwidth.

When the normal Workbench screen is displayed, there is usually enough time for each device to do its job without interfering with the others. But when the system is displaying a high-resolution screen with 8 or 16 colors, and an overscan width of 700 or more pixels, system resources are stretched a bit thin, even without a DMA hard drive making additional demands. In such situations, DMA hard drive interfaces are likely to slow down significantly, and possibly even develop timing problems.

Two things most people expect from DMA hard drive interfaces are speed and concurrency. But just because an interface uses DMA transfers doesn't mean that it is faster, or better integrated into the system than one that doesn't. A lot depends on the driver software.

For example, in our first hard drive review, we stated that non-DMA drives tended to affect multi-tasking, as seen by the jerky mouse while the drive was being accessed, and only one program at a time being able to access the drive (one DIR command would have to wait until the other had completely finished before starting its listing).

However, neither of the two non-DMA drives tested here exhibited those symptoms, while some mouse jerkiness was evident with the Pacific Peripherals DMA controller (though the company claims it is remedied in the latest version of the software). It is best, therefore, to be guided by a hard drive system's performance, rather than by buzzwords. It makes no more sense to base your hard drive buying decision on the letters DMA than it would to base your computer buying decision on the letters IBM.

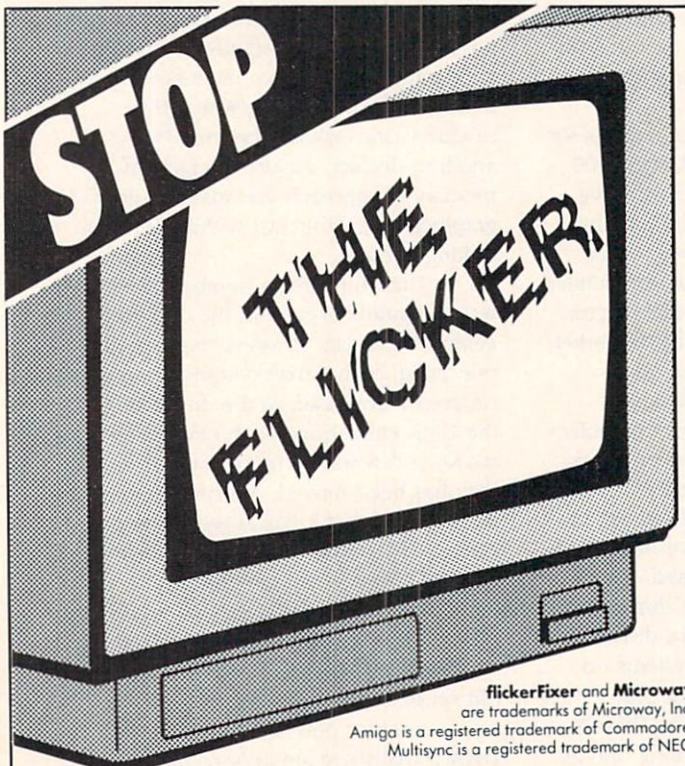
About the Tests

Each interface was installed in the same Amiga 2000 with the same 63 megabyte hard drive. The drive was formatted each time, and partitioned into three logical drives of 20, 33, and 10 megabytes each. The partitions were partially filled using the LV Backup program to restore several megabytes worth of files. The system was run with each interface for at least a week, to make sure that no serious problems arose in normal operation.

Several benchmark tests were run using each interface. The first was the ever-popular "Diskperf" program that we used in the previous hard drive comparison. This program times various operations, including creating and deleting files, scanning directories for file entries, moving the read head (seeking), and reading a block of data. It also measures the speed at which the drive reads and writes data blocks of varying size (from .5 Kilobytes to 32 Kilobytes). It also measured the total time it took for the Diskperf program to run; to provide an overall indication of how quickly the drive performed the tasks. The results of the Diskperf test are shown in Table 1.

Running Diskperf by itself doesn't indicate how a drive interface will perform in a multi-tasking environment. Therefore, other tests were devised. The first measures the drive performance under "heavy DMA" conditions.

For this test, two programs were run in the background while Diskperf was run. The first program displayed a 704 x 464 overscan picture with 16 colors. The second program, Sonix Play, played a 4-voice music score. Although this is not a typical example of how most people use their Amiga, it isn't really



flickerFixer and **Microway** are trademarks of Microway, Inc.
Amiga is a registered trademark of Commodore.
Multisync is a registered trademark of NEC.

flickerFixer eliminates your Amiga 2000's interlace flicker and visible scan lines. The result: **superior quality color or monochrome graphics and text** — for a full range of demanding applications, including CAD, desktop publishing, graphics, animation, and video.

flickerFixer fits into the Amiga video slot, is fully compatible with all user software, and does not modify the standard Amiga video signals. The board also upgrades the Amiga 2000 with a flicker free 4096 color palette, has an overscan mode that features a screen size of 704 x 470 pixels and drives most of the popular PC Multiscan and VGA monitors, including the NEC Multisync and Mitsubishi XC1429C.

Reviews are impressive: *Amiga GURU 6/88: "The display is fantastic . . . It is the best display we have ever seen on any computer system."* *AMIGAWORLD 5/88: "Our writers loved it . . . Compare the difference between night and day!"*

flickerFixer™

Advanced Graphics Adapter For The AMIGA® 2000

flickerFixer is priced at $595. It is made in the USA by Microway, Inc. For more information or to order, call Microway Sales at (508) 746-7341 or your Amiga Dealer.

Microway

P.O. Box 79
Kingston, MA 02364 USA
(508) 746-7341

32 High St.,
Kings-Upon-Thames, U.K.
01-541-5466

**NOW FCC
CLASS B APPROVED**

extreme. Some of the flashier video presentation programs combine large overscan displays with music; and the presentations created with these programs often require a large amount of data to be read in almost continuously off of the disk. The results given by the Diskperf program when run with sound and graphics in the background are shown in Table 2.

A third Diskperf test was designed to show how processor-intensive tasks affected disk performance, and vice-versa. For this test, a variant of Perry Kivolowitz's RAMspeed program was run continuously in the background while Diskperf executed. This program tests the speed of expansion RAM versus Chip RAM by timing a large number of processor memory move instructions. Since this program continuously uses the processor (without waiting for user input), it shows how the drive interface handles competition for processor time. And since it times a fixed number of operations, it also can be used to see how the drive activity affects a concurrently running application program. The results of the Diskperf tests, and the Ramspeed timings, are shown in Table 3.

Since benchmark programs like Diskperf tend to test artificial situations, we also performed a series of tests designed to correspond more closely to the operations real users routinely perform. The first test measured the time it took to copy an entire disk full of files to and from the hard drives. The disk in question was the Omega 6 version of Workbench 1.3, with 178 files in 21 directories, some 714 Kbytes in total.

The next test measured the loading time for loading an 84626-byte high-res overscan picture into Deluxe Paint II. This test was something of a ringer, as it was designed to demonstrate an interesting quirk of the Commodore 2090A driver. Since DPaint has no timing facilities, this timing was done with a stopwatch (the other timings were done with the same "Time" program used in the previous review). The final test measured the time it took to load and execute a 100,000 byte dummy program (100K of data and an RTS instruction).

All tests were run using the Fast File System from a pre-release version Omega 6 of Workbench 1.3. Although Fast File was not quite released at test time, it is so much faster than the old file system that it seems unlikely many hard drive owners will stick with the original system once the new one becomes available. Under these circumstances, we felt that testing the interfaces with the old file system would render this comparison obsolete as soon as it was printed.

Since the original round of hard drive comparisons included other devices such as the df0: floppy drive and the RAM: device, we decided to include the newest virtual drive, RAD:, the recoverable RAM disk from Workbench 1.3. Note that the times shown were for a RAD: drive formatted using the Fast File System. That's right, to use FFS on RAD:, you must first format the RAM drive. The speed difference is fairly dramatic. Using the old file system, RAD: is slower than most of the hard drives tested here. Using the FFS, it's quite a bit quicker.

Anomalies, Inconsistencies, and Mysterious Occurrences

Considering the preliminary nature of the FFS, and the relative newness of 2000 hard drive systems, surprisingly few problems were encountered. A couple of boards seemed to function inconsistently when used with the Micron 2 meg expansion board, but only when the FastMemFirst program was run before the drive was mounted. Great Valley and Supra are both making minor hardware changes to eliminate this problem. We also ran into trouble trying to run the JDK Images's Pro Video Plus program from the hard drive. None of the interfaces could successfully load the program more than 50% of the time. No problems were encountered with loading and running any other programs.

Test Summary

As expected, the DMA controllers (Commodore 2090, Pacific Peripherals Overdrive, and Supra A2000 interface) showed somewhat faster results from the Diskperf program than the programmed I/O controllers (Great Valley Products Impact card, and C Ltd.), particularly in the large buffer reads. A look at the overall time Diskperf took to execute in each case shows that the differences are not all that great. The Supra and Commodore 2090A interfaces turned in the best times, but the processor-controlled C Ltd. and GVP boards took only 18-23% longer to complete the task. Pacific Peripherals' Overdrive board took the longest, probably due to its rather slow writes.

When it came to multi-tasking, however, the results were much different. With graphics and sound running

Wilkommen

Bienvenue

Welcome

Magasins d'ordinateurs à l'Etranger et Marchands de Magazines

Bienvenue au monde stupéfiant de Amazing Computing™ - un des magazines les plus importants pour les utilisateurs du Commodore Amiga aux Etats-Unis. Amazing Computing™ fut le premier magazine à documenter CLI, le premier magazine ayant développé le projet d'extension mémoire 1 Mega pour Amiga , et le premier magazine à offrir une assistance serieuse de programmation.

Ne refusez pas à cette large partie de votre clientèle anglophone, ce magazine qui leur fournira la plus complète information sur l'Amiga. Venez et devenez membre de notre monde stupéfiant comme marchand ou annonceur.

Pour devenir un Amazing Dealer ou un Amazing Advertiser, nous vous prions de vous mettre en contact avec:

Marie A. Raymond
Coordinatrice
Internationale
PiM Publications
P.O. Box 869
Fall River, MA 02722
(508) 678-4200
(800) 345-3360
FAX: (508) 675-6002

Amazing COMPUTING

Your Original AMIGA Monthly Resource

Templicity™

55 Practical Applications
for Your Spreadsheet!

Available for Analyze!, Logistik, MaxiPlan, and VIP Professional.

- Try it for 30 days on our no-risk money-back guarantee.
- Increase control of your finances.
- Very easy to use.
- Ready-made templates save time.
- Adapted for your program.
- Flexible and affordable.

Over 50 worksheets for personal finance, home office, small business, real estate decisions, etc. in one package. Specify which spreadsheet program and disk drive you have.

Only $24.95 (before Jan. 1, 1989)

To order, send check/money order to:

The Sterling Connection
Box 4519
Berkeley, CA 94704

Or order by phone:
(415) 655-2355

Mastercard and Visa accepted.
California res. add $1.75 sales tax.

concurrently, the Commodore 2090A drive became much slower than any of the others. Although the two other DMA cards, the Overdrive and Supra interfaces, didn't slow down anywhere near as much, both caused the music player program to play the song slower than normal. Also, the Supra unit caused the overscan display to "tear" momentarily, several times during the course of the test. No processor-controlled C Ltd. or GVP board exhibited these problems. The Great Valley Impact card clearly came out the best in this test: it took twice as long to complete the Diskperf program as in the previous test. The others took anywhere from 2 1/2 to 7 times as long.

The results of the processor-intensive test also fall into line with our observations about DMA and processor-controlled interfaces. The DMA boards were hit hardest during short reads. In the 1/2K buffer test, both the 2090A and Supra cards went from 60 Kbyte/second reads to about 8 Kbytes/second. In the larger buffer tests, however, the DMA cards slowed down much less, and slowed the RAMspeed task down much less as well. The processor-controlled GVP interface was much less affected during short reads, but slowed the other task down more, particularly during the large reads and writes.

The miscellaneous tests in Table 4 put these other tests into perspective. As the other tests show, there is a maximum 20% speed difference between the various controllers for tasks like copying files. Thus, while the fastest controllers loaded a 100,000 byte file in a bit over 7/10s of a second, the slowest took a shade over 1 second. Not exactly what you'd call a monumental gap, particularly when you consider that the fastest of these boards are nearly half as quick as the RAM drive.

Note that the results given here are by no means cast in stone. All of the hard drive interfaces tested here are fairly new, and changes to the driver software are as inevitable as death and taxes. Such changes can cause big variations in the test results. An early version of the Supra software read a maximum of 50 Kbytes per second; the latest version's top speed is almost 6 times faster.

Conclusion

The good news is that all hard drive systems tested performed much better than those in our first test. Not only were they much faster (thanks to the new file system), but they performed more reliably as well. Not once in over 2 months of testing did I get the dreaded "Drive dh0: has a read/write error" or "Not a DOS disk in unit 1" requesters. The previous generation of drives were much more fragile, and prone to failure in the course of normal operation. The improved reliability of the drives and the availability of good backup programs like LV Backup means that the prudent Amiga owner need no longer fear catastrophic data loss should he switch to a hard drive.

Picking any one of these products, however, is a complex task. Even if you take the rather simplistic approach of picking the fastest drive interface, the question becomes, fastest under what conditions? Though the Diskperf results in Table 1 suggest that the Commodore 2090A is the fastest controller, the results in Table 2 shows that under certain conditions, it's the slowest.

And speed is by no means the only factor on which to base your decision. Price, features, and manufacturer support must be considered. Compatibility is another very important issue. Though most manufacturers test as many configurations as they can think of, there are bound to be some combinations of memory boards, processor accelerators and emulator boards that they haven't tested. This makes it important to buy your hard drive system from a dealer who will make sure it works properly with your equipment, or exchange it for one that will.

Despite the difficulty, if pressed to make a decision, I'd probably pick either the Supra or Great Valley Products Impact board. The Supra interface is one of the fastest tested, and was among the least affected by the DMA and processor tests. Its price is among the lowest.

The installation software supplied with the board makes the drive easier to set up than any of the others (though most of the companies are in the process of enhancing their installation software). The board has sockets for autoboot ROMs, so it should be easy to upgrade once 1.3 Kickstart ROMs become available. The most significant negative factor was the minor quirk mentioned in the discussion of the graphics and sound test. Considering the newness of the product, it seems likely that such problems will be ironed out when the software undergoes fine tuning.

The Great Valley SCSI interface was by no means the fastest one tested, but it was the most consistent. Because its performance figures changed so little during the graphics and sound tests, this board would be a good choice for those who do a lot of presentation graphics and video work. Like the Supra card, it includes the ROM sockets needed for autobooting. Great Valley makes this interface available in a number of convenient formats—expansion RAM/interface combinations, hardcards, and a 500 hard drive/expansion RAM combination. Some of the hardcard configurations should be quick, since they use fast Quantum and Conner drive mechanisms.

In the end, though, you'll have to make your own decision. There are a number of reasons why your choice might not agree with mine. Some people won't mind paying for the warm, fuzzy feeling they get from buying a genuine Commodore product for their 2000. Others will like the convenience of mounting their own 3 1/2" hard drive directly to the Overdrive board to custom-make a hard card. And for those who plan to buy the Dest scanner or

Laser Xpress printer, the C Ltd. board would be the natural choice for a hard drive interface. Whichever you select, rest assured that you'll be getting your money's worth. With a fast hard drive in your system, you will be constantly reminded of what a powerful personal computer the Amiga really is.

The Commodore 2090A Hard Disk Controller

The 2090A is the updated version of the 2090, the original DMA hard drive controller for the Amiga 2000. The only real difference between it and the 2090 is that the 2090A has a couple of sockets for the ROMs needed to autoboot the system from the hard drive (provided, of course, that you have the 1.3 Kickstart ROMs). Though the 'A' version of the controller was not yet available when this review was written, it is in production, and should soon become the official Commodore hard drive interface product. Those inclined to purchase a Commodore controller should note that at present, Commodore has no plans to retro-fit an auto-booting capability onto the earlier 2090 controller. Owners of the 2090 card may be able to add autobooting capabilities through third party products, however. Both M.A.S.T and Michigan Software are planning to market static RAM modules that you can boot from, for example.

The 2090A is unique among the controllers tested in that both SCSI and ST-506 interfaces come standard on the board. It allows you to connect up to two IBM-style drives and seven SCSI devices up to your Amiga 2000. This much flexibility doesn't come cheap, however. The 2090A lists for $399, more than any of the other interfaces tested. For purposes of comparison with the other SCSI controllers, only the SCSI interface was tested.

The 2090A comes on a full-sized A2000 card, about 4 1/2" high. The component side of the controller card includes a 50-pin SCSI connector, and two ST-506 drive connectors. The mounting bracket on the edge of the board carries a 25-pin connector for external SCSI devices. Physical installation of the hard drive system requires you to mount the drive in either a 3 1/2" or 5 1/4" drive bay, plug the power cable into the drive, connect the drive to the interface via a 50-pin ribbon cable (not included with the 2090A), and plug the drive into one of the 2000's 100-pin slots.

Software installation is relatively simple. The disk that comes with the controller hardware contains an installation program that can be run from an icon. This program takes you through low-level formatting, and installs the software needed to boot the drive on a copy of your Workbench disk. After the installation program finishes, you must run the AmigaDOS FORMAT command from the CLI before you can store any information on the drive. The entire installation process is well documented in the manual.

Although the installation program that comes with the controller is fairly good, it has a number of significant limitations. The program's format is more like the old-style PC applications than Amiga applications. It asks you a number of questions, one after the other, and has you type in the answers, rather than the normal Amiga procedure of setting up text gadgets and allowing you to fill each one. This means if you make a mistake, you can't go back to the previous question. Also, the installation program asks you to choose the drive type from a list which does not contain several of the more popular SCSI drives. For drives like the Seagate 277N, you must fill in the number of heads, tracks, and sectors per track. While this information usually comes with the drive, it isn't the kind of thing the average user knows.

Nor does the software help much with partitioning the drive. The installation software sets up a default partition (DH2: for SCSI drives), but if you want to divide the drive into additional partitions, you must edit the DEVS:Mountlist file. Here, the information on the number of heads and sectors per track is critical—if you don't know it, the drive won't format. Since DH2: becomes the boot partition when autobooting is enabled, it cannot be formatted with the Fast File System. Therefore, if you use the FFS on the 2090A, you'll have to learn how to partition the drive.

Although I have the 1.3 Kickstart ROMs, the 2090A didn't actually boot my system from the hard drive automatically. That's because the 277N drive takes so long to warm up that when the controller looked for the drive, it wasn't ready yet, and the "Workbench 1.3" prompt screen appeared. A warm reset (CTRL-AMIGA-AMIGA) allowed the boot process to succeed, however, and later versions of the boot ROM will probably may wait longer before timing-out. It should be

Wilkommen

Bienvenue

Welcome

All Foreign Computer Stores / Magazine Dealers

Welcome to the amazing world of Amazing Computing™ - one of America's leading magazines for Commodore Amiga users. Amazing Computing™ was the first magazine to document CLI, the first magazine with a 1 Meg Amiga upgrade hardware project, and the first magazine to offer serious programming assistance.

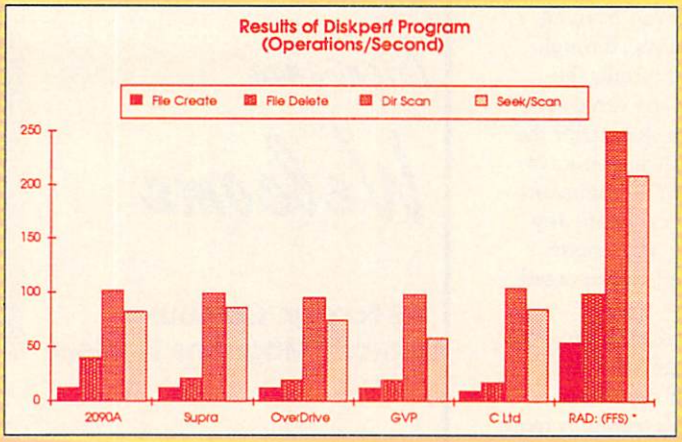
Don't deny your large English-speaking audience the kind of magazine that will provide them with complete information on the Amiga. Come and join our Amazing world as either a dealer or an advertiser.

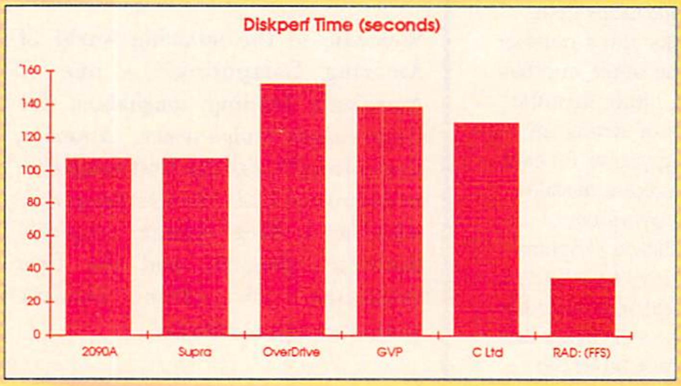
For details on becoming an Amazing Dealer or an Amazing Advertiser, please contact:

Marie A. Raymond
International Coordinator
PiM Publications
P.O. Box 869
Fall River, MA 02722
(508) 678-4200
(800) 345-3360
FAX: (508) 675-6002

Amazing COMPUTING
Your Original AMIGA Monthly Resource

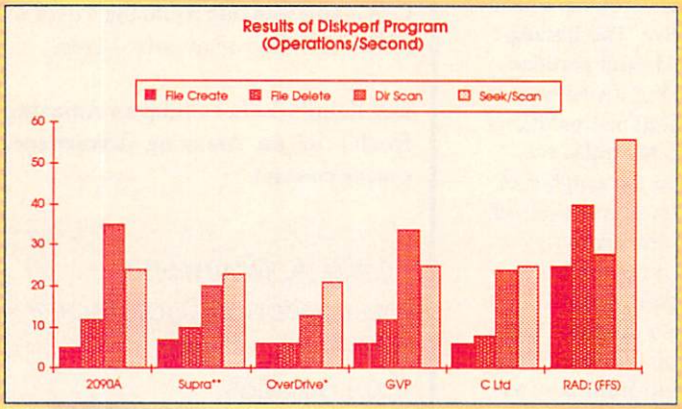
Table One: Results of Diskperf program (with no other programs running concurrently)

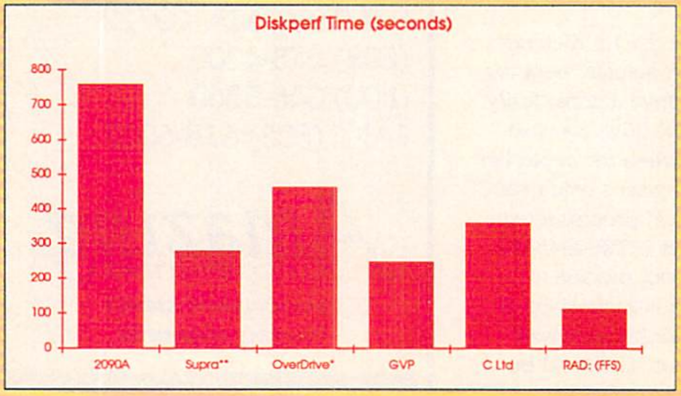




	2090A	Supra	OverDrive	GVP	C Ltd	RAD: (FFS)
File Create	12/sec	13/sec	12/sec	13/sec	10/sec	55/sec
File Delete	41/sec	22/sec	20/sec	20/sec	17/sec	100/sec
Dir Scan	102/sec	100/sec	96/sec	98/sec	104/sec	1100/sec
Seek/reads	83/sec	87/sec	75/sec	59/sec	86/sec	209/sec
512 byte buffer						
Read (Kb/sec)	55.6	62.4	27.8	28.7	58.2	160.0
Write (Kb/sec)	27.5	26.9	26.7	27.5	41.3	150.6
4K buffer						
Read (Kb/sec)	106.7	142.2	142.2	159.6	116.4	640.0
Write (Kb/sec)	134.7	127.9	58.1	88.2	69.2	426.7
8K buffer						
Read (Kb/sec)	182.8	213.3	213.3	159.6	142.2	853.3
Write (Kb/sec)	150.6	160.0	56.8	111.3	64.0	512.0
32K buffer						
Read (Kb/sec)	320.0	284.4	256.0	170.6	170.6	1280.0
Write (Kb/sec)	232.7	196.9	60.9	121.9	71.1	640.0
Diskperf Time (seconds)	107.3	107.2	152.2	138.8	129.1	34.9

Table Two: Results of Diskperf program with 4-voice Sonix Play song and 704 x 464 x 4 overscan display running concurrently





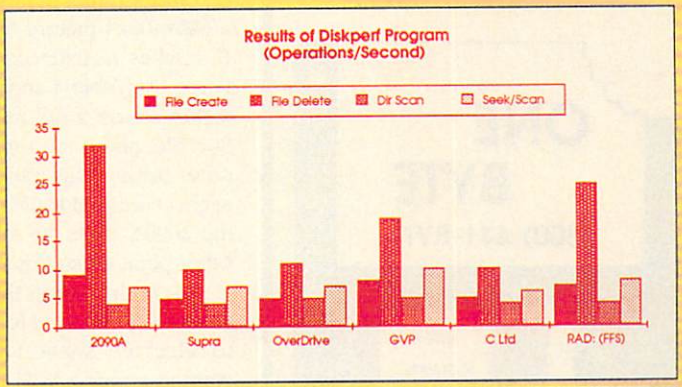
	2090A	Supra**	OverDrive*	GVP	C Ltd	RAD: (FFS)
File Create	5/sec	7/sec	6/sec	6/sec	6/sec	25/sec
File Delete	12/sec	10/sec	6/sec	12/sec	8/sec	40/sec
Dir Scan	35/sec	20/sec	13/sec	34/sec	24/sec	28/sec
Seek/reads	24/sec	23/sec	21/sec	25/sec	25/sec	56/sec
512 byte buffer						
Read (Kb/sec)	20.3	17.9	12.3	17.9	20.8	44.1
Write (Kb/sec)	14.9	13.9	11.4	16.6	17.8	35.0
4K buffer						
Read (Kb/sec)	28.1	62.4	23.7	77.6	39.4	256.0
Write (Kb/sec)	36.1	49.2	20.6	67.4	28.4	196.9
8K buffer						
Read (Kb/sec)	19.5	106.6	27.2	106.7	44.1	284.4
Write (Kb/sec)	27.2	71.1	23.8	75.3	31.6	196.9
32K buffer						
Read (Kb/sec)	4.1	196.9	49.2	134.7	38.2	512.0
Write (Kb/sec)	3.1	106.6	35.0	102.3	22.1	284.4
Diskperf Time (seconds)	761.9	277.9	462.2	248.9	358.3	113.8

* Sonix Play song slowed down during disk activity

** Sonix Play song slowed down, and momentary screen glitches were noted during disk activity

Table Three: Results of Diskperf program with Ramspeed program running concurrently

	2090A	Supra	OverDrive	GVP	C Ltd	RAD: (FFS)
File Create	9/sec	5/sec	5/sec	8/sec	5/sec	7/sec
File Delete	32/sec	10/sec	11/sec	19/sec	10/sec	25/sec
Dir Scan	4/sec	4/sec	5/sec	5/sec	4/sec	4/sec
Seek/reads	7/sec	7/sec	7/sec	10/sec	6/sec	8/sec
512 byte buffer						
Read (Kb/sec)	8.0	7.3	8.2	18.5	6.9	8.0
Write (Kb/sec)	17.5	6.8	7.8	18.3	6.6	8.8
4K buffer						
Read (Kb/sec)	67.4	54.5	58.2	116.4	50.2	64.0
Write (Kb/sec)	102.4	48.3	36.0	64.0	37.6	77.6
8K buffer						
Read (Kb/sec)	121.9	91.4	85.3	85.6	77.6	121.9
Write (Kb/sec)	106.7	82.6	40.0	71.1	45.7	116.4
32K buffer						
Read (Kb/sec)	284.4	196.9	160.0	134.7	150.6	512.0
Write (Kb/sec)	182.9	160.0	53.3	106.7	65.6	320.0
Average Ramspeed Time (in seconds)						
(run with diskperf)	64.2	69.2	77.4	77.4	69.9	60.6
(run alone)	59.1	57.4	57.8	57.8	57.2	57.2


Results of Diskperf Program (Operations/Second)

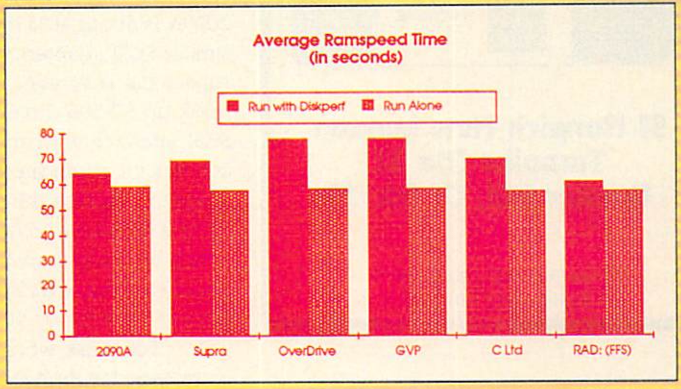
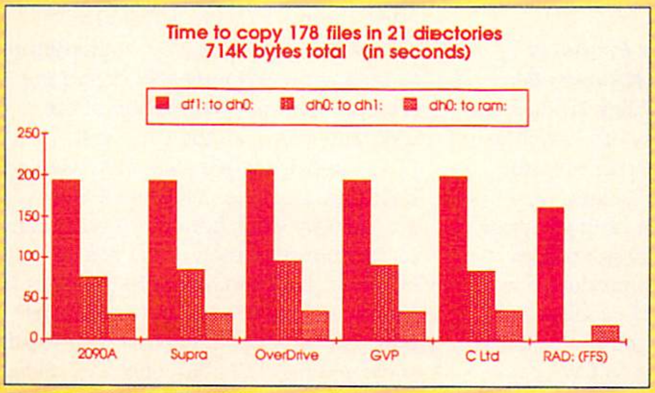
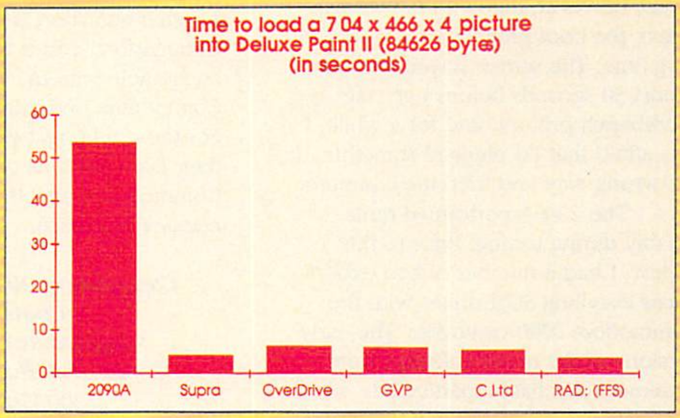
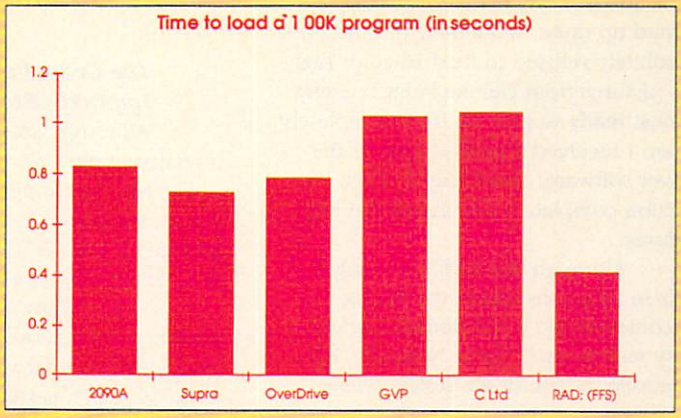

Average Ramspeed Time (in seconds)

Table Four: Miscellaneous Tests

	2090A	Supra	OverDrive	GVP	C Ltd	RAD: (FFS)
Time to copy 178 files in 21 directories 714K bytes total (in seconds)						
df1: to dh0:	194.5	195.5	207.3	196.6	201.7	164.8
dh0: to dh1:	76.7	86.8	97.7	92.2	86.9	——
dh0: to ram:	31.4	31.9	36.2	35.8	37.0	19.1
Time to load a 704 x 466 x4 picture into Deluxe Paint II (84626 bytes) (in seconds)	53.66	4.5	6.5	6.3	5.6	3.8
Time to load 100K program (seconds)	0.83	0.73	0.79	1.03	1.05	0.42


Time to copy 178 files in 21 diectories 714K bytes total (in seconds)


Time to load a 7 04 x 466 x 4 picture into Deluxe Paint II (84626 bytes) (in seconds)


Time to load a 1 00K program (in seconds)

We take a byte out of the price
Not out of your pocket!

ONE
BYTE
(800) 441-BYTE

51 Norwich New London
Turnpike Rte 32
Quaker Hill, CT 06375

Amiga is a registered name of
Commodore Amiga, Inc.

noted, however, that with the version tested, the boot process took quite a long time. The screen stayed white for almost 30 seconds before I got the Workbench prompt, and for a while, I was afraid that I'd plugged something in the wrong way and fried the computer.

The 2090A performed quite reliably during testing. Prior to this review, I had a number of bad experiences installing SCSI drives with the Commodore 2090 controller. The early versions of the driver software were extremely unreliable, particularly when the controller was used with the faster SCSI drives. Read/write error messages turned up quite frequently, and the drive absolutely refused to load 16-color high-res pictures from Deluxe Paint II. I was almost ready to give up on it completely, when I received a new version of the driver software, dated May 5. This version completely eliminated the read failures.

Although the 2090A no longer fails in situations where the timing becomes tricky, it still doesn't perform very well in such cases. Normally, this controller is one of the fastest ones tested. But when a high-res, 16 color screen is displayed, the transfer rate drops to 15% of normal, or less.

While the new version of the driver no longer fails when trying to load a 640x400x4 picture from Deluxe Paint II, it takes an extraordinarily long time to do so. As Table 4 shows, it took the 2090A almost a full minute to load an 80K file, about ten times longer than the other drives. This particular problem seems confined to DPaint II, however; the 2090A loads the same picture in other paint programs with no trouble.

At this point, there aren't many compelling reasons for the average user to select the 2090A. For those who only want to connect SCSI hard drives, the 2090A is about 40% more expensive than similar SCSI adapters, and offers no real substantial advantage. Users who want to hook up ST-506 drives can buy a Supra SCSI interface with an ST-506 to SCSI adapter for about the same price. Such adapters improve the performance of the ST-506 drives by providing a RAM buffer which increases transfer speeds, and RLL data encoding, which increases the storage capacity by 50%.

For those who don't mind a bit of extra expense, however, the 2090A is about the fastest controller available, at least in the Workbench environment. Another attraction is that it's made by Commodore, and is therefore likely to be compatible with the widest range of Commodore and third-party peripherals. For those who rest easier knowing that their 2000 contains only genuine Commodore parts, the 2090A is a reasonable choice.

Commodore 2090A Hard Disk Controller
Suggested Retail Price $400
Commodore Business Machines
1200 Wilson Drive
West Chester, PA 19380

The Great Valley Products Impact SCSI/RAM Controller

Although Great Valley Products is a relatively new company, it's not unfamiliar with the Amiga market. Gerard Bucas, the company's president, was formerly a Vice President of Engineering at Commodore's West Chester offices.

Great Valley's first product is a unique combination RAM expansion/SCSI interface board for the Amiga 2000. Originally, the board was to have come populated with 1 megabyte of memory in 256K RAM chips. Unfortunately for GVP (and just about everybody else), the introduction of their board coincided with a shortage of 256K RAM chips that forced them to re-think their strategy. They now offer the controller in two flavors, one that holds up to 2 megabytes of the more cost-efficient megabit chips (256x4), and one that holds up to 1 megabyte using 256x1 chips. Both boards are sold without memory chips, so that you can add the memory yourself, either now, or later, when the price of RAM (hopefully) drops. We tested the 2 megabyte version, with no memory installed.

The Impact board is solidly built and nicely finished (the ends of the edge connector have been beveled for easier insertion, and the 25-pin external SCSI connector is riveted, as well as soldered, to the board). The sixteen sockets for RAM chips line up against the top edge of the board, and two jumpers are used to set the RAM configuration. The card also has two sockets for autoboot ROMs, though the ROMs were not available for testing at the time of the review. A 50-pin SCSI cable is included for connection to an internal drive.

The board comes with both a printed manual and an installation disk. The manual is extremely thorough, covering the mechanical installation of the controller and the hard drive, the software initialization process, RAM installation, multiple SCSI devices, and setting up various hard disk partitions. The manual contains a lot of nice extras like a description of the elements of a Mountlist entry, and pin-outs of both the 25 and 50-pin SCSI connectors. It is also is the only one I've seen that warns about shutting down a drive too soon after drive activity. This is a warning well worth heeding, since a delayed write may occur a few seconds after a longer write has finished. If you turn off the power before all write activity has ceased, the results may be as disasterous as pulling out a floppy disk before the red light goes off—corruption of the disk, and loss of all the data on it.

Initializing the disk and creating a boot disk are done using script files, activated by icons from the Workbench. In the current version of the software, however, the Devs:Mountlist file is set up for one or two 20 megabyte partitions only. Any other configuration will require some editing of the Mountlist file. Setup follows the normal AmigaDOS procedure. Do a low-level format (if desired), run the BINDDRIVERS program, the MOUNT program, and the FORMAT program to prepare the drive for use by AmigaDOS. The Great Valley board is the only other besides Commmodore's own to store the hard disk driver program in the Expansion drawer, and use BINDDRIVERS to load it. In the future, Great Valley plans to include a comprehensive installation program with an Intuition interface, that allows you to interactively set/initialize your drives.

Advertisements for the Impact board refer to it as "DMA SCSI", but that description is somewhat misleading. Company officials explained that the board "DMAs into the on-board memory cache". In plain English, that means the card has it's own memory buffer, which is used to store information as it comes off the drive, before being passed to the Amiga via "processor-controlled I/O". While the on-board cache undoubtedly speeds things up, the term "DMA" is more commonly understood to mean that the controller moves information directly into the Amiga's memory space, not into its own memory buffer.

Although not the fastest controller in all situations, the Impact board's performance was quite respectable overall. As explained in the overview section, DMA transfer has costs as well as benefits, and the GVP board was obviously designed to avoid the penalties incurred by DMA controllers when competing for scarce system resources. While it was about 25% slower than the DMA controllers under normal Workbench conditions, it proved faster, on the whole, than all of the other controllers tested with high-res graphics and sound running in the background. This consistent level of performance makes the board a natural choice for those who use the Amiga to run elaborate graphics and sound presentations. In addition, the single-slot RAM expansion and SCSI controller should appeal to those who wish to leave some room in their 2000 for future expansion.

GVP Impact SCSI/RAM Controller
Suggested Retail Price: $360 (2 meg capacity, 0K installed)
325 (1 meg capacity, 0K installed)
Great Valley Products
P.O. Box 391
Malvern, PA 19355
(800) 782-1841

Pacific Peripherals Overdrive Board

The OverDrive board from Pacific Peripherals was the first SCSI-only DMA alternative to Commodore's 2090 to appear for the Amiga 2000. Although it occupies a full-size board that's the same size as the 2090A, the components are all packed into the left side of the board. The bare right side of the board has four holes that can be used to mount a 3 1/2" hard drive directly to the controller card. A hard drive power cable is also provided on the right side of the board for this purpose. The Overdrive was the only one of the interfaces tested that was designed specifically for use as a hard drive card, keeping both of the extra drive bays free for other drives.

The current version of the OverDrive has no provisions for autobooting under Kickstart 1.3; there aren't any sockets for boot ROMs on the board. Pacific Peripherals has indicated, however, that it plans to make a daughtercard available at the time Kickstart 1.3 ROMs are released, that will add autobooting capabilities. The price of the upgrade has not been announced.

All of the manuals for setting up the Overdrive are on disk. The information in the files that comprise the manual is fairly complete, but the files themselves don't have any icons, which means that they can only be read from the CLI. The only printed documentation is a one-page summary card which reviews the software installation procedure. The information on this card, along with the auto-executing Startup-Sequence on the disk, should be sufficient for most users to install the software. Partitioning the drive, or installing the Fast File System, however, will require some work with a text editor.

Unlike most products tested here, the OverDrive comes with an installation program that uses an Intuition interface. The ODUtils program uses a series of pull-down menus to allow you to select and format various drives. The drive type selection is performed automatically, by issuing a SCSI command to the drive asking it to identify itself.

The program will also let you manually select the drive type. After you've selected the drive, another menu item lets you perform the low-level format procedure. When that's completed, you may check the drive for bad blocks, and map out any that are found. A final menu item has the program add

Wilkommen

Bienvenue

Welcome

Alle ausländische Computergeschäfte / Zeitschrifthändler

Wilkommen in der erstaunlichen Welt von Amazing Computing™ - eine von Amerikas führenden Zeitschriften für Commodore Amiga Benützer. Amazing Computing™ war die erste Zeitschrift, die über das CLI berichtete, die erste Zeitschrift mit 1 Meg Amiga Hardware Projekt Verbesserung und die erste Zeitschrift, die seriöse Programmhilfsassistenz anbot.

Wenn Sie mehr Information über die Stelle des Amazing Verkäufers oder Amazing Werbefachsmanns haben möchten, kontaktieren Sie bitte:

Marie A. Raymond
International Coordinator
PiM Publications
P.O. Box 869
Fall River, MA 02722
(508) 678-4200
(800) 345-3360
FAX: (508) 675-6002

Amazing COMPUTING™

Your Original AMIGA Monthly Resource

The Tshell makes using AmigaDOS fun. It makes the interface between you and the Amiga free of complications, intuitive and easy, while at the same time allowing complete control over the Amiga's operating system.

Although it is a **professional tool** having capabilities of an interpreted programming language, the Tshell is simple to learn and use. Even if you're not a programmer, you can use the Tshell immediately as a high-powered CLI replacement.

While closely resembling the popular UNIX command shells, the Tshell is freshly designed and developed for the Amiga. As a result, it is an **advanced command interpreter** that is an **improvement** upon UNIX. Here are some of the Tshell's features:

Commands 40 UNIX-like builtin commands offer efficiency, speed and features from the Tshell environment; ARP commands; extra utilities; full AmigaDOS compatability; background and detached processes; AmigaDOS residency; builtin commands can be RAM or disk resident; default 'cd' command. **Editing environment** line editing and command history: insert, delete, kill, and put lines; autoindent; read/write files; edit/re-execute partial history; definable function keys and dynamic prompt. **On-Line Help** two levels; customizable/extensible; quick help for reminder of command usage; full help displays command manual page or manual chapter (ARP command documentation included too). **Files** file name expansion with standard UNIX matchings, directory/file differentiation and directory tree searching; current, parent and previous directory shorthand. **I/O redirection** Standard redirections plus write append; standard error stream for builtin commands; pipes; here documents; backquoted commands. **Variables** string/integer shell variables; environment variables; indirected variables and variable/string concatenation. **Operators and Expressions** Simple numbers, numeric variables and return (exit) values can be used in expressions as 32-bit integers. All C language unary and binary operators, including assignment operators, are supported. **Statements** List processing 'forall', and C-like 'for', 'while' and 'do/while' looping statements; C-like 'if/else' (incl. 'else if') and 'switch/case' (with pattern matching) conditional statements. **Procedures** Redefine existing commands; write your own Tshell commands and programs; multi-line alias procedures for builtin (RAM resident) execution; shell scripts for execution from disk files; procedures accept parameters and return exit values like compiled commands. **Interprocess Communication** ARexx-compatible with user-definable message part name; send and receive command messages.

There's more! Small ad, great product. The Tshell is an extensive, flexible and highly configurable system. You'll love it!

Get it right now! Send a check or money order for $49.95 to
Metran Technology, Box 890, West Oneonta, NY 13861

NY residents PLEASE SEE YOUR DEALER. The Tshell has been shipping since February 1988. We ship immediately on receipt of order. Shipping/handling charges are included. Send $2.00 and SASE for freely-distributable demo disk or just ask for our flyer for more descriptive info. Phone: (607)432-4836.

UNIX, Amiga and Tshell are trademarks of AT&T, Commodore-Amiga and Metran Technology, respectively.

the proper entry to the DEVS:Mountlist file. After exiting the program, use the AmigaDOS FORMAT command to ready the drive for data storage.

The performance of the controller during the tests was generally good, though write speeds were somewhat slower than the other DMA controllers. This was partially a result of the 2:1 interleave setting which was chosen in order to optimize read speeds. Other interleave settings might provide faster writes. Though the Overdrive was not quite as slow as the 2090A in the graphics and sound test, it did slow down the Sonix Play score that was playing in the background.

The OverDrive has enjoyed a position as the only low-price DMA alternative to the Commodore 2090 for a number of months. Now, however, it faces some serious competition from the Supra controller, which uses DMA transfers, has an autobooting capability built in, and is priced the same as the OverDrive. The only area in which the Overdrive appears to retain an edge is in its hard card design. At least until Microbotics comes out with its Hard Frame controller, the OverDrive remains the only controller board specifically designed for custom-building your own hard disk on a card.

Overdrive Controller
List Price $250
Pacific Peripherals
P.O. Box 14575
Fremont, CA 94539
(415) 651-1905

The C Ltd. SCSI Host/Controller

C Ltd. was one of the first companies to release a SCSI adapter for the Amiga, and today it carries the widest line of SCSI Amiga products. They sell complete hard drive systems for the 500, 1000, and 2000, ranging in capacity from 20 megabytes to over 700 megabytes. They have lots of experience with more exotic storage devices as well, including tape backup systems, 10 megabyte removable floppies, WORM drives, and CD ROMs. They've worked on a networking system based on their SCSI hardware. And soon, the company will expand into the desktop publishing market with Laser Xpress, a SCSI laser printer, and the Dest optical scanner, which connects via a SCSI interface. In addition to its direct marketing effort, C Ltd. also licenses its controller hardware and software to other drive manufacturers, including Pioneer and Phoenix.

The newest of the company's SCSI interfaces is its A2000 controller card, which is the least expensive SCSI adapter available for the Amiga. Looking at it, it's easy to see why. While a controller like the Commodore 2090A crams almost 50 chips onto a large circuit board, the processor-controlled (non-DMA) C Ltd. interface uses exactly seven chips and a handful of discrete components. It's less than three inches high, and it's small size makes it a bit harder to plug in than the other boards, as well as less mechanically stable. The unit we received for testing didn't even have a mounting bracket to attach it to the frame. In addition to the standard 50-pin internal and 25-pin external SCSI connectors, the board also has a connector for the front panel LED used to indicate drive activity. There is no provision for boot ROMs, and C Ltd. has no plans to add autobooting. They plan to take a "wait and see" approach toward such an upgrade.

The rear part of the board has space where an Omti ST-506 to SCSI adapter can be mounted. Many of the drive configurations C Ltd. sells use such an adapter. Pioneer Computing uses the same space for a mounting bracket for a 3 1/2" hard drive, turning the controller into a hard drive card. As sold by C Ltd., however, the board is not really meant to be used as a hard card.

C Ltd. sells most of its SCSI adapters as part of complete hard drive systems, and ships their hard drives preformatted, with a lot of useful public domain software already on the disk. The installation software provided with the SCSI controller, therefore, is not very sophisticated. It consists of an AmigaDOS batch file used to run the programs needed to prepare the drive for use. As of this writing, installation instructions were all on disk—the latest version of a printed manual was not ready.

For the test, I used a pre-release copy of the Version 3 software. This software is designed to accommodate a wide range of SCSI devices other than hard drives, as well as networking over the SCSI port. It requires the user to create a text file called Devsetup as well as the Mountlist file. Devsetup is used by a program called DevInstall that is run before the hard drive is mounted. Although C Ltd. hasn't come up with much of a user interface to set up a drive using this software, all of the tools are there, including utilities to do low-level formatting and mapping out bad blocks on ST-506 drives (SCSI drives automatically map out bad blocks).

The company does plan, however, to write a mouse-driven installation program that will write the Devsetup and Mountlist file for the user. Aside from the DevInstall program, drive installation and mounting follows the standard Amiga procedures. The MOUNT command is used to let DOS know the drive is there, and the FORMAT command is used to prepare it for storage.

Although the driver software was a pre-release version, the drive behaved quite reliably during the test period. While it was among the slowest tested, there are a number of extenuating circumstances to consider. The C Ltd. interface seems particularly sensitive to the drive interleave settings. Different settings gave wildly different results on the Diskperf test.

At lower interleave settings, the reads were faster, but the writes slower. The 32K buffer Diskperf test showed a maximum of 213 Kbytes per second on reads, and 48 Kbytes per second on writes. At higher interleave settings, reads peaked out at 128 Kbytes per second, while writes increased to 111 Kbytes per second. The 3:1 interleave setting we chose for our tests was a compromise value, but clearly, you can optimize your drive for either reads or writes using this interface. Optimizing for reads should yield somewhat better results than the benchmarks indicate. Another factor to consider is that C Ltd. sells many of its systems with fast ST-506 drives and an Omti ST-506 to SCSI adapter. This combination is reported to be significantly quicker than the standard SCSI configuration tested here.

The C Ltd. controller has a lot going for it. It's a time-tested design that's backed with a full one year warranty. It's versatile enough to handle a wide variety of peripherals, from WORM drives to laser printers. But most of all, it's cheap. It's inexpensive design allows companies like Pioneer Computing to offer complete 30 megabyte hard drive systems for as low as $600. Though the speed of such systems might not be world-class, they're still quite respectable, and you can't argue with the price.

SCSI Host/Controller
Suggested Price: $200 (A2000 version)
$250 (A500 version)
$300 (A1000 version)
C Ltd.
723 East Skinner
Wichita, KS 67211
(316) 267-3807

The SupraDrive A2000 Interface

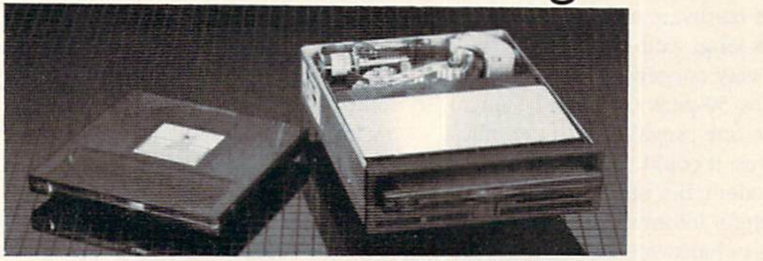
Supra Corporation is another company whose hard drive products for the Amiga have been around long enough to have been in our first round of reviews. Supra has offered complete hard drive systems for the 1000 and 500 for quite a while, but now they've added an A2000 controller to their lineup. Unlike previous SupraDrives, this new interface card does DMA data transfers.

The board contains three mounting holes which can be used to attach an ST-506 to SCSI adapter (Supra sells some versions of the board with this adapter already attached). Although these holes could be used to mount a 3 1/2" drive directly to the board in order to make a hard card, they weren't intended for this, and Supra doesn't encourage this practice.

PIONEER COMPUTING Does it again!

PRD-44

44 Megabyte
Removable Cartridge
Winchester Disk Drive



The PRD-44 Cartridge Hard Disk is an innovative design that incorporates reliable Winchester Technology in a removable media. You will never out grow this Winchester as the PRD-44 offers you unlimited storage. The PRD-44 has many benefits, including low cost, mass storage, back-up capability, transportable and rugged cartridge media.

At 44 megabytes per cartridge, individuals may maintain large amounts of data for individual or share system applications. The compact 5¼" cartridge permits data security as users may remove and secure sensitive data.

The drive has an average access time of 25 milliseconds and a 1:1 interleave capability. Within 20 seconds after cartridge insertion, the PRD-44 has completed spin-up, self-test diagnostics and is ready to accept commands. Additionally, automatic error detection/correction and extensive defect management are transparent to the user.

KEY FEATURES:

- Removable Cartridge Media
- 44 Megabyte (formatted) Storage Capacity, per Cartridge
- Unlimited off line storage
- Fully Integrated SCSI Controller
- 25 Millisecond Average Access Time
- Standard Half-Height 5¼" Size
- 20,000 Hour MTBF
- Data Transfer Rates up to 1.25 Megabytes per Second
- 8 K Data Buffer
- Read/Write Operations with 1:1 Interleave
- ECC and Automatic Retry
- Transparent Defect Management with Track and Sector Sparing
- Self-Diagnostics at Power Up
- Rugged Hard Disk Cartridges

PRD-44 44 Megabyte Removable Cartridge Disk Drive
$999.99
PRD-44c 44 Megabyte Cartridge
$129.99

Take advantage of our package deal:

Disk Drive & Cartridge
$1049.99
Disk Drive, Cartridge & SCSI Controller
$1279.99
Disk Drive, Cartridge,
SCSI Controller & Case
$1399.99

PIONEER COMPUTING
2469 East 7000 South #200 - Salt Lake City, Ut 84121
(801) 942-1174
ORDER DESK 1-800-999-3013

COMPUTER OUTLET, INC.
(800) 544-3665
In CT 872-2667

SOFTWARE • ACCESSORIES
Aegis' PORTS OF CALL
Retail $49.95 - You pay $35.99
Phoenix 3.5" Ext. Drive - $189.95
Cinemawares' ROCKET RANGER
Retail $49.95 - You pay $35.99

Call for prices on Aegis - Mindscape - Activision
Epyx - E.A. - Discovery - all others
Amiga - 64 - PC Compatibles - Apple - Mac - Atari

Visit our retail store at
335 Talcottville Road
Vernon, CT 06066

Call for complete price list
NEVER ANY FREIGHT CHARGES *
No charge card surcharges
The price you see is the price you pay!

Check - M/O - M/C - Visa - Discover - Shipping U.P.S. Ground *Orders under $25.00 add $2.00 handling. C.O.D. orders under $100.00 add $2.20 U.P.S. C.O.D. fees. CT residents add 7.5% sales tax. Manufacturer's warranties apply. Not responsible for machine compatibility. 30 day exchange policy on defective software. Prices subject to change W/O notice. We cannot sell Commodore Amiga brand hardware thru the mail.

Supra provides the most complete installation kit of any of the packages tested. The hardware installation manual is 18 pages long, with 30 photographs covering every conceivable aspect of the process. The 50-page Operator's Manual is the same one provided with the other Supra Drives; it could use updating for the new system, but otherwise, provides a lot of helpful information. There's even a little bag of hardware that includes a SCSI cable, a power cable extension, wires for the drive LED, a SupraDrive sticker, and the standoffs and screws needed for mounting an internal drive. This makes the Supra system the only one that you can install without a making a trip to the hardware store.

The Supra installation software is also outstanding. The SupraFormat program is the only package tested to allow the user to control every aspect of the initialization process from a single, mouse and menu-driven screen. Like the Overdrive utilities, SupraFormat queries the SCSI controller in order to automatically identify the drive type. If the drive can be identified, the controller and disk type are displayed, as well as the SCSI device number, drive interleave, the number of surfaces (heads), blocks per track, and cylinders. If the drive can't be identified, all of these parameters can be entered manually in the appropriate string gadgets. There are also spaces for entering landing zone, step rate, and write pre-compensation figures where needed (for non-SCSI drives).

Setting up drive partitions is a snap. You click on the partition name, type in the size in megabytes, and the program automatically calculates and displays the starting and ending cylinders, and the amount of storage space that remains to be allocated. You may also select the old or new file system for a partition by clicking a gadget.

When you allocate all of the storage space, you can format the drive or just zero the partitions (a quick format), mapping out bad blocks if you so choose. When the format procedure is finished, the partition information is written to the drive, eliminating the need for updating the Mountlist file. Then all you do to boot the drive is use the SupraMount command—the AmigaDOS formatting has already been done for you. Although the SupraFormat process is much simpler, Supra still allows you to use the AmigaDOS setup procedures if desired. Simply set all of the partitions to zero, zero them, and use the AmigaDOS MOUNT and FORMAT commands.

The SupraDrive controller that was tested included a pre-release version of the autoboot ROMs. Running a special Convert program activates a special partition, BD0:, which takes up all of the cylinders below the low cylinder of DH0: (future versions of SupraFormat will include the option to perform this function). This partition, which is automatically formatted with the old file system, becomes the boot drive. As with the 2090A, Supra's pre-release autoboot ROMs didn't wait long enough for the 277 to come up to speed. Resetting once the drive warmed up allowed the system to boot from the hard drive, however.

In the Diskperf and file copy tests, the Supra controller ran neck-and-neck with the 2090A and o loaded the DPaint picture and the dummy program in the shortest time. Although drive transfers didn't slow down anywhere near as much as with the 2090A during the graphics and sound test, the Supra driver did slow down the Sonix Play scoreduring the test, and even caused some momentary screen glitches. To be fair, though, the version of the software that was tested could hardly be termed mature, as the tests were performed within a couple of weeks of the board's introduction. These flaws were the only problems that were noted, and can be eliminated with a little fine tuning.

Of all of the controllers tested, the Supra interface stands out as an exceptional value. It costs only fifty dollars more than the lowest priced controller, but includes top-of-the-line features such as autobooting, DMA transfer speeds, and a SCSI cable and mounting kit. Its instructions are among the clearest, and its installation software the easiest to use. Moreover, Supra is a solid company that's been selling a wide variety of peripherals in the Apple, Atari, and Commodore markets for a number of years. In a volatile industry like personal computers, that's a pretty good indication that they must be doing something right.

SupraDrive Controller
Suggested Retail Price $249
Supra Corporation
1133 Commercial Way
Albany, OR 97321
(503) 967-9075

•AC•

The WShell

by L. S. Lichtmann

Long ago I became convinced that one of the things Commodore did right was provide the parallel user interfaces of the Workbench and the CLI. Icon-based interfaces are wonderful for casual use and for novices, and command line interfaces remain unsurpassed for power and flexibility. Unfortunately, the deficiencies of the standard CLI environment and AmigaDOS are legion and too notorious to elaborate here. These problems are slowly being rectified by Commodore and by the vision of Amiga users and developers.

One of the most popular shareware enhancements for the CLI interface is ConMan, by William S. Hawes. ConMan adds command line editing and command history to the CLI. Anyone who has ever tried to get a listing for a directory four levels down can attest to the value of these additions. Now Mr. Hawes is back with WShell, a complete replacement for the CLI.

What the WShell Is Not

The computer term "shell" has two slightly different connotations. In its original use, a shell is a piece of system software responsible for mediating between the user and operating system, particularly in interpreting user input and loading and executing programs. However, especially in the MS-DOS world, shell has come to mean an applications program which simplifies the user's interaction with the computer. By presenting a graphic interface, a shell trades power for ease. Some shells (in this latter sense) and shell-like CLI enhancements are already available.

WShell, however, is a shell in the older sense. It provides the full power of a command line user interface by substituting for the standard CLI and providing complete access to programs in the "C:" directory with many more options for command entry.

The WShell

Unlike Mr. Hawes's previous program, WShell is a standard commercial product rather than shareware. Registered users of ConMan were offered an advance copy of WShell's release version at a discount. (Those of you who may have been using ConMan without having sent the author a payment, your sins have come home to roost!)

WShell runs on any Amiga, although it requires Version 1.2 of the system software. The program comes on a single half-full disk. The distribution disk has three major components: ConMan, the WShell files proper, and ARP (each of which I will discuss). The distribution disk is not a boot disk, and a number of files must be copied to your system disk to use the WShell. Mr. Hawes has thoughtfully provided batch files to perform the necessary copy operations. As you might expect from an author who has produced shareware programs, there is no copy protection on the distribution disk or in any of its programs.

WShell's documentation consists of a sixty-page typeset manual, spiral-bound so it lies flat on a desk. The organization is excellent, and the descriptions and explanations are uniformly clear. Novices should be warned, though, that familiarity with AmigaDOS and the CLI is assumed. The manual includes a good index, and WShell lists for $49.95.

ConMan

ConMan performs the same functions for WShell that it does when used with the standard CLI, primarily giving WShell line editing and command history capabilities. If you're not familiar with ConMan, it consists of four files: "conhandler" and "conhandler.library" (which must live in the "L:" and "LIBS:" directories of your system disk respectively), and "conman" and "setcman" (normally placed in your "C:" directory)

After ConMan and all desired options (typically in your startup-sequence) are activated by the "conman" command, all CLI or WShell windows subsequently activated are provided with ConMan's services. The "setcman" command allows modification of ConMan options once the program is running.

In addition to supporting basic editing functions—such as backing up over the command line to allow overwriting incorrect characters or inserting new ones—ConMan supports a full set of special editing characters. For instance, the Shift-Left Arrow and Shift-Right Arrow combinations back up or advance one word in the line. The Control-Y combination deletes to the end of the line. The F1 and F2 keys control window size: F1 toggles the active window between its current size and an "iconified" (minimum size) version; F2 toggles the active window between its current size and a full screen version.

Command history is accessible through the Up Arrow and Down Arrow keys. Previously entered commands are individually recalled from a buffer (most recent first) by successive pushes of the Up Arrow. Successive pushes of the Down Arrow move one command in the other direction through commands already recalled with Up Arrow.

The new version of ConMan (V1.1) which is on the WShell disk has added command history search functions. Typing a line of text and pressing the F5 key searches the history buffer for the last command that has the entered text as its start. Searches in the other direction (toward more recent commands) are performed with F6 key. Version 1.1 also has a utility command "history" that can be used to display the entire command history, save the

NEW FROM THE

computer club™

Help You Can Rely On!!!

the

AmigaDOS

CLI Reference Book

ISBN 0-9621268-0-2

- 8 Pages Long
- CLI Command Usage Guide
- Lists Standard Devices and Assignments
- Lists All *ED* Editor Commands
- The First In a Series Of Amiga Reference Cards
- *GREAT LOW PRICE!* ($3.95 US)

Send your order to:

The Computer Club Company
4131 Meadow Hill Lane
Fairfax, VA 22033-3113

(Checks or Money Orders Accepted)

For Large Quantity Purchase Information Call: (703) 968-7588

Amiga, AmigaDOS, and The Computer Club are trademarks of their companies.

command history to a file, or to load a new command history from disk. This feature allows you to pre-load sets of frequently-used commands.

WShell Activation

WShell needs ConMan to function. Along with the ConMan files, an additional file, "wshell.library," must be moved to the "LIBS:" directory. The distribution disk contains an icon—analogous to the CLI icon supplied on the Workbench disk—to activate the WShell. WShell can also be activated from a command line using the "NewWSH" command which functions like AmigaDOS's "NewCLI" command. The NewWSH command, however, allows you to specify either an initial command for immediate execution by the new WShell or a sequence of commands found in a specified disk file.

This arrangement forms the basis of my new startup-sequence file:

```
conman -c     ; activate ConMan

; Startup a WShell with the specified console window
;   and execute the commands in the file Boot-WShell

NewWSH con:/30/640/170/Boot_WShell/c from s/Boot-WShell

endcli >nil:      ; now close the startup CLI
```

The commands I formerly had in my startup-sequence file now reside in "Boot-WShell"."This way they are executed from a WShell and have complete access to WShell facilities. New WShell-specific commands can be executed at boot-up time. A sample of this sort of "split startup" procedure is supplied on the WShell disk. The WShell has an additional startup feature: if neither an initial command nor a specified file is supplied, the new WShell looks in the "S:" directory for a file named "Startup-WShell." If it exists, WShell executes the commands in the file before issuing its first prompt.

Built-in & Resident Commands

A standard complaint about AmigaDOS is that it is entirely disk-based. Unlike MS-DOS, AmigaDOS has no "internal" commands. The case for having an "external", disk-based system can be and has been argued persuasively. Nevertheless, I have frequently found myself wishing that a number of commands had been made internal. The WShell attempts to improve on AmigaDOS in this matter in two ways: "built-in" commands and "resident" commands.

Built-in commands are crucial to WShell, but you can delete the corresponding command files in the "C:" directory to save disk space. The built-in commands are mostly batch file commands (such as "if" and "echo") that perform elementary functions. Users are unlikely to replace these with their own versions. "Endcli" is also a built-in command, but you should retain the "endcli" command on disk for terminating any non-WShell command line interfaces you generate.

"CD" and "prompt" are also built-ins. Prompt is particularly interesting because it recognizes an extended set of special characters. As well as permitting insertion of "%N" (current task number) into a prompt string, the WShell's prompt command also allows insertion of the current directory, time-of-day, error return codes, or any of about ten other special strings into the prompt. As well as a "standard" CD command, the WShell has an "implicit" CD command—you can type a recognizable path and directory name without any command name, and the WShell assumes you want to do a CD to this directory and perform the change.

Some built-in commands do not correspond to any AmigaDOS commands. For instance, the "pushcd," "popcd," and "swapcd" commands allow you to maintain a stack of frequently used directories, facilitating quick moves around a disk's directory structure.

Alternatively, you can define your own set of internal commands with the "resi" (resident) command. For instance, typing

```
resi dir
```

loads the "dir" command into memory from disk and adds its name to WShell's list of resident commands. Subsequent "dir" commands can be executed without reading the command from disk. WShell keeps statistics on the use of the commands in the resident list.

Giving the "resi" command with command line options lets you view the

Zorro II Prototyping Board

- Over 4400 Plated Holes on a 0.1" Grid.
- Gold Plated Edge Connector.
- "D"-type I/O Connector Pattern.
- Accepts 64 Pin DIPs and 14x14 PGAs.
- Low Inductance Power and Ground Pattern for High-Speed Designs.
- Designed for Maximum Flexibility.
- Includes Mounting Bracket.

To order, send:
check or money order for $49.95 +
local sales tax (California only) +
shipping & handling (US: $3.00, Foreign: $6.00)
in US dollars to:

Celestial Systems
Department M
2175 Agate Court
Simi Valley, CA 93065-1839
(805) 582-0729

list of resident commands, reorder the list based on frequency used to minimize search times, or remove resident commands from the list. WShell's resident commands capability is preferable to storing commands in a RAM disk because resident commands take less memory, less time to execute, and do not require a special search path to find commands on disk and in a RAM disk.

Resident commands are not restricted to things like "dir" and "assign." Programs normally thought of as application software—text editors, for instance—can be made resident commands if they are written properly.

WShell searches the list of resident commands before searching the list of built-in commands. So if there is some command you *must* use instead of a WShell built-in, you can do it. By prefacing the command name with the left square bracket character ([), you can also force WShell to bypass the search for resident and built-in commands and go straight to the usual search path.

ARP

There is one restriction on using resident commands, and it sets me up for a digression on the third major element of WShell. Resident commands must be "pure" code. The command must not modify its own code as it executes—any "scratchpad" memory must be external to the command. (The other term used for such code is "re-entrant.") Pure code is desirable for multi-tasking systems because it allows several independent tasks to share one loaded copy of a program, making memory use efficient.

Unfortunately, most programs are not pure code; not even all the AmigaDOS "C:" directory commands are pure code. The WShell distribution disk comes to the rescue in the latter case, however. WShell comes with the latest version of the ARP (AmigaDOS Replacement Project) commands, public domain replacements for Commodore's "C:" directory AmigaDOS commands.

The ARP commands are faster than their AmigaDOS counterparts, much smaller (freeing a great deal of space on your Workbench disks), and have a more consistent set of command line parameters. Many have extra capabilities not found in the corresponding AmigaDOS commands, and a few new commands not present in AmigaDOS at all are included. Equally important for the WShell, all the ARP commands are pure code. (In fact, ARP has a "resident" command which serves the same function as WShell's "resi," although I prefer "resi" because it has more capabilities.)

New Store!

Fisher's
Computers & Software

Authorized Amiga Dealer

8005 Archibald
Rancho Cucamonga, CA 91730
(714)987-1662

Serving
The Inland Impire

Aliases

WShell also provides "aliases" as another way of customizing the Amiga's command set. Aliases define multiple names or abbreviations for commands. If you are more comfortable with a certain trademarked operating system directory command ("ls", for instance), but still want to give the Amiga's "dir" command, give the WShell the command

```
alias ls=dir
```

Thereafter, typing "ls" executes "dir." The "dir" command file is not renamed, so it is still accessible by entering "dir." The "alias" command also supports multi-word replacement. Entering

```
alias print="type >prt:"
```

activates the command "print file1" and sends the contents of file1 to the printer. Alias can also cause WShell to recognize particular strings as abbreviations for standard commands. The case of the characters in the alias string specify characters that must always be typed for the command. Thus

```
alias MKdir=makedir
```

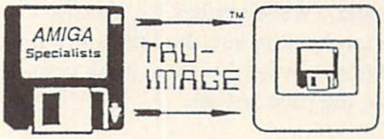
causes "mk," "mkd," "mkdi," and "mkdir" to be recognized as "makedir." Other options allow you to list aliases currently in effect, or to delete particular aliases.

Pipes

Pipes chain together commands so the output of one command serves as the input to the next without speed and space penalties associated with setting up an intermediate file. You can set up pipes by inserting the character "|" between commands you want to chain. For example, a program called "Tee" takes any input given to it and copies that input to both the screen and the "standard output" (which is also the screen unless otherwise specified). The command line then

```
dir | Tee >dirfile
```

35MM SLIDES FROM YOUR ARTWORK!



Professional 35mm Slides
- Now you can have reproduction and presentation quality slides of your work
- Distortion-free--fills in raster lines crisp bright colors, converts all IFF files

Now Custom graphic art and illustration.

$10 each for your 1st to 4th slides.
5 to 9 slides-$8.50
Over 10 slides-$8.00
Add $2.00 for shipping.
New York residents add sales tax.
Call (212) 777-7609 FOR DETAILS
Ask for Ilene--or write TRU-IMAGE
P.O. Box 660, Cooper Station
New York, N.Y. 10276
Amiga is a trademark of Commodore-Amiga.

causes a directory listing to be sent to the program "Tee." The redirection causes the output to be temporarily redefined as a disk file named "dirfile." Thus, this command line produces a directory listing on the screen which is also saved to a file—a function long on my wish list!

To use pipes, you must mount a "pipe handler." The mounting operation may be unfamiliar to many users. You must have a file named "mountlist" in the "DEVS:" directory of your system disk containing information needed by the operating system. An appropriate "mountlist" file for the pipe handler is provided on the WShell disk. You must also issue the command

```
mount PIP:
```

before you can use pipes (usually in your startup-sequence). The WShell manual includes clear instructions for this procedure, although it would have been nice if the WShell installation batch file had set up an appropriate mountlist file.

The WShell Environment

WShell provides itself one form of "environment variables,"global values that set certain modes of behavior. To set up the WShell environment, you create a special directory (on a permanently mounted device, such as the RAM disk) and assign the device ENV: to this directory. The environment variables are the files in this directory, and the values of the variables are the text contents of the files. For example, the WShell looks for an environment variable named "titlebar." If you have a file in the "ENV:" directory named "titlebar," and it contains a valid WShell prompt string (the arguments of a WShell "prompt" command), WShell writes the specified prompt string into the title bar of the WShell's window. Thus, if the contents of the file "titlebar" are

```
%C
```

WShell writes the current directory into the titlebar and updates the bar every time a "cd" is executed.

In addition to "titlebar," the current version of WShell uses other environment variables. "Path" sets a global search used by all WShell windows as a "last resort" when hunting a command. "Echo" controls display of alias translations. "Shellwindow" controls the default size, etc., of new WShell windows.

Even Up The Score!

INVESTOR'S ADVANTAGE

Let your Amiga give you the Advantage in making *better investment decisions!*

Color graphics of Individual Stocks and General Market Trends help you make more profit in this volatile market. High Low Close, Moving Averages, Centered Moving Averages, Volume, Relative Strength, Stochastics, Wilder's RSI, Cycles, Trend lines and Momentum. Powerful reports such as the Relative Strength Report help you pick the best performers. Use the Market Barometers to help you time your market entries. Update Stocks, Mutual Funds and Commodities manually or automatically. Easy to use communications included.

Only **$99.95**

See your local Dealer or Call:
Software Advantage Consulting Corporation
37346 Charter Oaks Blvd
Mt. Clemens, MI 48043 (313) 463-4995

Amiga and the Investor's Advantage are trademarks of their respective companies.

The ARexx Connection

One of Mr. Hawes avowed purposes in developing the WShell was to integrate his ARexx language (available as a separate product) into a command-line environment. ARexx is an Amiga implementation of the REXX language that originated on IBM mainframes. ARexx really deserves a review of its own, but I can summarize by saying that it is an interpreted language (like BASIC) that runs on a memory-resident interpreter loaded at boot time.

The language's features make it useful as a powerful alternative to the usual AmigaDOS EXECUTE batch files, and as a medium for writing additional high-level commands. The "Tee" program mentioned above is an ARexx demonstration program included on the WShell disk.

ARexx also defines a standard interface that makes it possible for application programs to easily add macro facilities and for different application programs to talk to one another. (Some commercial programs, such as Microsmith's TxEd Plus, already have ARexx interfaces.)

WShell provides transparent support for ARexx. To run an ARexx program once the interpreter is running, simply type the name of the program. No special command, such as the AmigaDOS "EXECUTE," is necessary. WShell also recognizes in-line ARexx programs if the appropriate ARexx statements are entered between double quotes.

Final Remarks

Taste in software is a notorious idiosyncrasy (particularly in user interfaces), and recommendations should be filtered through your own likes and dislikes. I really like WShell. I tried to think of criticisms, ways in which I felt it was deficient, and came up blank. The software performs very useful functions and works as advertised. The documentation is excellent, and the package is an outstanding value even at full list price. All in all, I think the WShell is a shining example of what commercial software should and can be. I can't imagine how I got along before I had it.

•AC•

ROOMERS

by the Bandito

[The statements and projections presented in "Roomers" are rumors in the purest sense. The bits of information are gathered by a third party source from whispers inside the industry. At press time, they remain unconfirmed and are printed for entertainment value only. Accordingly, the staff and associates of Amazing Computing cannot be held responsible for the reports made in this column.]

Hardware news: While the RAM shortage is far from over, things are getting better. Prices are dropping, though they are still far above what they used to be. Availability is also improving. Once some of the new chip factories "RAMp" up, perhaps RAM costs will drop the way they did in the good old days.

The RAM shortage has already had an impact on the hardware market, though, with some product prices increasing and others kept from the market entirely. For instance, Progressive's FrameGrabber went up from $499 to $599 because of the high cost of RAM chips. As the Bandito speculated, NewTek's Video Toaster will be delayed because of the RAM shortage. Now they're saying Christmas, but they promise the product will be better for the delay. Let's hope they can stick to their original price —the way RAM chips have gone up, that would be some trick. The Bandito expects a higher price tag and a later ship date.

In this fast-paced industry, it seems that companies are always changing the roster. This time it's Progressive Peripherals, who have ended their cross-marketing relationship with Precision Software from England. Two employees left Progressive to join Precision, which will now market its products directly in the U.S.

Also at Progressive, Dan Browning and Eric Stein departed, and Pixmate programmer Justin McCormick is now in charge of marketing. (The Bandito thinks more programmers should be involved with marketing. Too many marketing beings barely know how to turn on a computer.)

MicroIllusions has lost one of its hot programmers. Reichart, author of FirePower and other titles, is forming his own development group. He claims to have an unbelievably large number of programmers working with him. Currently in development is a BBS-based role playing game for the Amiga with some hot graphics. Reichart has also designed an Amiga animation program to help professional animators.

More Micro-I news: Jim Steinert, head of MicroIllusions, is trying to sell KJ Computers, the store where he started the company. It seems that trying to run a retail store and produce software at the same time is just too much of a hassle.

Our friend Ami is starting to get bigger movie roles — A1000's appeared in Disney's recent Sunday Movie *Not Quite Human.* Some Amiga graphics were even shown. Wire-frame animation was shown; ray-traced Amiga graphics would have looked too good to be convincing! Both the good guys and the bad guys had Amigas, which just shows that Amigas are not inherently good or evil. (Remember, Amigas don't kill people— people kill people) If graphics are outlawed, only outlaws will use graphics.

What does Commodore have against Ami-Expo? Well, Deep C (the Bandito's Commodore connection) states that Commodore was upset when Ami-Expo tried to get the right to put on the Amiga Developer's Conference. Of course, that may not have anything to do with the fact that Commodore is trying to get developer support for the World of Commodore show, instead of for Ami-Expo. Developers will probably support both, but with less enthusiasm for World of Commodore since it's not Amiga-specific. Who wants to hang around with a bunch of C64's, anyway? The future is Amiga. This is the last Christmas for the C64...but then again the Bandito has heard that line for the last three years.

In other Ami-Expo news, the LA Amiga Users considered boycotting Ami-Expo because of its high price. At press time, Ami-Expo had not yet happened, so the Bandito doesn't know if they'll go through with the threat. It costs $25 to get in, plus $15 for hotel parking, which adds up to a pretty penny. That's the price of a good piece of software — hmmm, maybe they should give a program away at the door. Any corporate volunteers out there?

Game news: Starglider II is shipping, and it has the fastest frame rate of any real-time 3D game.(The Bandito knows there's a whole audience out there who just cares about frame rate—to heck with the game!) Rocket Ranger is shipping, and sales are red-hot. Players say that Cinemaware has more game in there along with the graphics and sound.

News you won't hear anywhere else: Though Apple and Commodore are deadly rivals in the marketplace, that doesn't prevent them from doing business together. The Bandito has learned that the Apple IIGS's 15-voice music chip is made by none other than MOS Technologies—which is owned by Commodore. Makes you wonder if the next Amiga might not have a few more voices on the sound chip, eh?

Commodore expects to sell 300,000 Amigas this Christmas. Developers are crossing their fingers and clutching their prayer beads. If Commodore is right, low-priced, hot-selling items could be the big winners; they make perfect Christmas gifts. This scenario could help out a number of

Now On Videocassette!

How To Use The AmigA For Video Productions

How To Get Started!
How to Hook Up Your System!
How To use Genlock!
Plus Video Software Info.

Only $39.95
For VHS

Send Check or Money Order To:
TV One Productions
1027 Sussex Ave.
Deptford, NJ 08096
Call
(609) 848-5698

Include $2.00 For S & H
Canada Add $4.00 For S & H

smaller developers. Or it could line the pockets of big developers who can afford to charge less. The Bandito expects $15 games to be hot items. Are they really selling 100,000 Amigas a month right now? The Bandito hopes so, but that seems like an awfully high number.

HAM Paint Wars: the latest dispatch from the front is that Deluxe PhotoLab is doing okay, especially among power users, but isn't flying off the shelves like they thought it would. With no advertising, what do they expect? DPaint III should be coming soon—perhaps by Christmas—though it won't add HAM capabilities. It does have some animation tools, and every DPaint owner will want to upgrade.

What happened to Digi-Paint II and Photon Paint II? Both were supposed to be out in the spring, and it's beginning to look like fall around here. Whispers are that both are adding features and squashing insects. The Bandito thinks both will try hard for a Christmas ship date—whether or not they'll make it is debatable.

Just to set the record straight for the less informed gossip columnists, Steve Jobs *already* owns Pixar. By the way, Pixar *is* developing rendering software and a "Pixar-on-a-card" for the Mac II. That would be a nice Amiga product, but don't hold your breath.

More Steve Jobs news: The NeXT computer will finally appear October 12. It's taking some pages from Amiga's book—great stereo sound, a graphics coprocessor for fast animation, zillions of colors, and high resolution. However, it's going to be very expensive—RAM prices have hit here, too. The operating system is a UNIX variant, and they've licensed Display Postscript from Adobe for the screen.

The NeXT computer is supposedly just for the higher education market, but don't believe it. Jobs wants the whole enchilada—he's after the Macintosh and the PS/2 market share, as well as the Sun workstation market. This guy thinks big. In fact, he's just inked an agreement with his old arch-rival IBM, giving IBM the right to use NeXT's display technology (interface). This deal gives Jobs more money and a great deal of prestige in the business world.

Interestingly, Apple has reviewed the NeXT computer and signed off on it, saying that it does not infringe on any proprietary Apple technology (as per Jobs' agreement that he signed when he left Apple).

Atari is suing the former owners of the Federated Group, claiming that Federated misrepresented the company's value when they sold to Atari. Since Atari is still losing gobs of money at Federated, this is one way to try to make up for that. At the same time, Atari has let it slip that they wouldn't mind selling Federated, but no interested buyers have yet appeared. In other news, Atari raided a number of New York dealers and seized $100,000 worth of counterfeit Atari 2600 machines and cartridges. Gee, wonder why the Taiwanese pirates haven't bothered to counterfeit Atari ST's?

What is Epyx really doing? The I-VCR thing the Bandito described previously was apparently misinformation from an Epyx *apparatchik*. Now the Bandito hears that it may be a 16-bit videogame. After all, Amiga originally made a joystick, just like Epyx does now, and then produced a computer you may be familiar with.

There are also quite a few Amiga alumni at Epyx these days. They can't be daring enough to take on the expensive fight-to-the-death videogame market, can they? They sure are. Darker rumors hint that the machine is based on a 68000, and that they've licensed the Amiga chip set for graphics and sound. (Many people are interested in getting rights to the Amiga's technology, from what the Bandito hears.) They're obviously trying to capitalize on the availability of 68000 programs and programmers to rapidly create a large base of available software.

What we may have here is an Amiga truly configured as the ultimate home videogame with a target price around $250. The A500 may already have that market, though, even with its higher price. It seems to the Bandito that other publishers might be wary of producing software for a machine owned by a competitor. That would give Epyx the money they need to improve its position in the software market.

Epyx's machine isn't the only 16-bit videogame threatening to appear. Nintendo is developing its 16-bit game machine. They released it in Japan and plan to bring it over here next year. Also, NEC has its PC Engine on the way that uses an oddball 16-bit processor and some cool 256 color graphics as a high-end game machine. They plan to offer it for about $200 with an optional CD-ROM drive for another $300 to $400.

Adding a CD-ROM player begins to make the PC Engine sound like a CD-I player, though at a lower price. And what is happening with Compact Disc - Interactive? As the Bandito reported earlier, it is running "way late." Now they're talking Xmas 1989 for the introduction of the first players. Prototype players should arrive at developers in a few weeks.

Developers are still spending incredible amounts of effort to develop for CD-I. One major publisher has already spent over a million dollars on CD-I development. Why so much interest in CD-I, even though it's so late to market? Well, publishers and developers talk about the TV quality graphics and CD quality sound, the animation, the amount of storage, and the real mass-market potential. But a lot of sexy technologies before have failed to attract to much attention. (The Amiga still doesn't get the respect it deserves from software developers.)

The Bandito thinks the reason for all the developer enthusiasm is more fundamental than graphics, sound, storage, or mass market. The basic fact is that CD-I products *can't be copied.* Publishers and developers both salivate at the thought of eliminating piracy. Of course, somebody will get around this limitation (and DAT or THOR-CD might be just the ticket), but until then, it's profit city. After all, publishers are convinced that for every legitimate copy someone actually pays for, anywhere from two to ten freebies are out there. If they could get all that lost revenue, they'd be rich (or so they think).

Speaking of Tandy's read/write optical disc, Tandy is staffing up with engineers to try and make THOR-CD a reality. It's likely that they won't have a commercial product for another two years or more, but they may mess up the marketplace until then as everyone waits for it. It's the old Jack Tramiel strategy—announce some vaporware and see how many orders you get, while the competition runs around chasing its tail trying to figure out if you really mean it.

One way to remedy CD-I's lack of writeable storage is to add an A/D converter and a little circuitry (about $6 in parts) so the CD-I players can record on your VCR. That would make plenty of storage, and it would enable some more common program types to use CD-I. (What good is a paint program unless you can save images)? Of course, DAT or THOR-CD could make this a moot point.

Great news for Commodore: Apple has just raised prices on most of its computers. The Mac II went up nearly 30%! And the IIGS went up nearly 10%, while extra memory for it went up 77%. This makes the distinction between the Mac II and the A2000 sharper than ever. Why did they do it? Well, Apple's DRAM contract expired. To meet demand, they must buy chips on the spot market, which eats into their (aptly named) gross profit margins. So they raised prices, figuring customers would flinch, but buy the stuff anyway. This means prices will stay up if demand does. Apple is happy to make the extra profit.

This also means the IIGS Plus will be more expensive. The Bandito hears that the IIGS Plus release is being delayed, partly because of the price increase. Apple will probably keep the decision up in the air until the last minute, but look for them to ship either in time for the late spring or late fall education market buying times.

Apple just introduced the IIc Plus, which has a 3.5 inch drive and a faster processor. List price is $1099 for a one-drive, color system. (It's amazing how fast those horse-drawn carriages can go, isn't it?) Apple also introduced the Mac IIx, with a 68030 instead of a 68020. For some reason, the press reports the speed is only about 15% faster—must be a slow 68030 chip.

The important news is the new 1.4 megabyte disk drive that can automatically read IBM 3.5 inch disks and provide software to help with some file conversions. Why can't Amiga do this? It's the only kind of compatibility most people need. Get some conversion software and build that drive into the next A2000. EA's DeluxePaint on the PC uses IFF files, so graphics shouldn't be a problem. Why fool around with bridgeboards...and the extra expense? Most people just want to read in word processing or spreadsheet files, anyway.

While we're on the subject of Other People's Computers, IBM's PS/2 line is under fire. While IBM claims great success for the PS/2's, most of the PS/2 computers they sell still have the old bus architecture. (Their MicroChannel has not been exciting) Most buyers look at it and think,"Gee, I can't use any of those expensive cards I bought for my old PC, and this new Micro Channel doesn't have any cards." So all the other MS-DOS clone makers have created a new bus standard that will use all the old cards, as well as new cards that offer faster performance. It will be interesting to see which side wins in the "bus battle." And we thought we had problems switching to the Zorro standard.

But that's not the only IBM news. As you may recall, they tried to get into the home and education markets with the ill-fated PC*jr* (with the infamous chicklet keys). They were laughed out of the market, nursing their wounds. They tried again with the PS/2 Model 25, but that got nowhere in home or education. Lack of sound was a major reason.

Well, IBM doesn't give up easily. Some developers have already been shown a new IBM model (or parts of it — IBM is very tricky when it comes to new stuff). The rumors whisper several things, but the version the Bandito likes is an 80286-based machine with VGA graphics and an ADPCM sound chip that has CD quality digitized sound, a blitter chip for fast animation. All for about $1500. Sounds like they want to knock off an Amiga, doesn't it? Look for something like this at Spring Comdex, and tell'em the Bandito told you.

•AC•

HIGHER PERFORMANCE...AND CHEAPER TO BOOT!

FData-10 Single 3.5" External Drive **$149.95**

FData-20 Dual 3.5" External Drive w/Power Supply **$299.95**

- Fully 1010 Compatible
- Ultra Compact Design
- Acoustically Quiet
- Amiga® Color Coordinated
- Ultra Low Power Consumption
- Daisy Chainable
- Extra Long Drive Cable
- High Performance
- No Annoying "Clicking" Noise
- Diskette "Dust Protection" Door
- Super Low Price

DISK PERFORMANCE SOFTWARE

ASDG FAC II	$29	**Central Coast Software** Quarterback	$55
Microsmiths, Inc. TxEd+	$55	**Central Coast Software** DOS-2-DOS	$39
Fuller Computer Project "D"	$39	**Discovery Software** Marauder II	$27
Progressive Peripherals Directory Master.	$39	**Sony** 3.5" DS/DD (Box of 10)	$19

POLICY: Shipping and handling extra. Personal checks require 3 weeks to clear. For faster service use credit card, cashier's check or bank money order. Credit cards are not charged until we ship. All prices are U.S.A. prices and are subject to change, and all items are subject to availability. These prices reflect a 5% cash discount. For all credit card purchases there will be an additional 5% charge. Defective software will be replaced with same item only. All sales are final and returned shipments are subject to a restocking fee.

FLEXIBLE DATA
SYSTEMS, INC.
10503 FOREST LN. • SUITE 148 • DALLAS, TX 75243

214-669-3999
FAX: 214-669-0021

Amiga® is a registered trademark of Commodore-Amiga, Inc.

VISA MasterCard AMERICAN EXPRESS Card

KEYCLICK

Program a typewriter click into your Amiga keys.

by Mike M. Duppong
Delphi: MOCKO GEnie: MMD

One thing I always missed on the Amiga was an audio click when keys were pressed. While the Amiga keyboard has the best feel of any I've typed on, it doesn't have the audio feedback that aids touch-typing. Thanks to the multitasking operating system, adding this feature is relatively painless for both user and programmer.

When you run the program you see a window that contains a slider to adjust the click volume. Each time you press a key or a mouse button, an audio click sounds. Simply close the window to end the program. That's about all users have to know.

For you programmers out there, the center of activity revolves around a custom input handler. The handler simply signals the waiting task when a key or mouse button event arrives in the input stream. Once the task is awakened, it sends a command to the audio device to initiate the click and goes back to sleep. Therefore, this program only uses a significant amount of processor time when a key or mouse button is actually pressed.

The program details are in the source code, so here I will comment only on its peculiarities. The program initially ran at priority zero— or, if you ran from CLI, the current CLI priority. This sometimes made the program sluggish, so, for better response, I let the program set its own priority to forty. You can change priority by assigning a different value to TASK_PRIORITY, defined at the beginning of the program.

In a busy system, I managed to confuse the audio device on several occasions by pressing keys as fast as I could. I could use some feedback on how to prevent this, but it shouldn't be a problem when you're typing normally—it takes some real keyboard-bashing to cause the error. The problem seems to be that the audio device clams up and won't talk to the program. An error message pops up on the screen's title bar when the Keyclick window is selected. To get around the problem, just close the current Keyclick window and run the program again.

LISTING ONE

```
/* Keyclick V1.0 by Mike M. Duppong  [3/29/88] */
/* Last modified: 6/7/88 */
/* Written in Aztec C V3.6A */
/* Compile: cc keyclick.c */
/* Link: ln -o keyclick.o keyclick */
/* You may contact me through Delphi (MOCKO) */
/* or GEnie (MMD) */

#include <exec/types.h>
#include <exec/memory.h>
#include <exec/ports.h>
#include <exec/errors.h>
#include <exec/devices.h>
#include <exec/execbase.h>
#include <intuition/intuition.h>
#include <devices/input.h>
#include <devices/inputevent.h>
#include <devices/audio.h>
#include <functions.h>

#define HANDLER_PRIORITY 51 /* Priority of input handler */
#define AUDIO_PRIORITY 127 /* Priority of click sound */
#define TASK_PRIORITY 40L /* Priority of the task itself */
#define WAVELENGTH 2L   /* # of bytes in square-wave */
#define PERIOD 600L /* Waveform period */
#define CYCLES 20L  /* Number of waveform cycles */
#define CLUP_INPDEV 1   /* Close-up flags for cleanup() */
#define CLUP_AUDDEV 2   /* Audio device */
#define CLUP_HANDLER 4  /* Remove handler */

struct IOStdReq *iostdreq;
struct IOAudio *ioaudio;
struct Interrupt interrupt;
struct MsgPort *input_port, *audio_port;
struct IntuitionBase *IntuitionBase;
struct Window *window;
struct PropInfo prop;
struct Image *image;
struct Gadget volgadget =
{
  NULL,                         /* NextGadget - none */
  26, 12,           /* LeftEdge, TopEdge */
  128, 8,           /* Width, Height */
  GADGHCOMP,                    /* Flags */
  GADGIMMEDIATE | RELVERIFY,/* Activation */
  PROPGADGET,                   /* GadgetType */
  NULL,                         /* GadgetRender- set below */
  NULL,                         /* SelectRender */
  NULL,                         /* GadgetText */
  NULL,                         /* MutualExclude */
  (APTR)&prop,                  /* SpecialInfo */
  NULL,                         /* GadgetID */
  NULL                          /* UserData */
};

struct NewWindow windowdef =
{
  0, 13, 178, 22,   /* X, Y, width, height */
  0, 1,                         /* Detail/block pen */
  CLOSEWINDOW | GADGETUP, /* IDCMP flags */
  NOCAREREFRESH | SMART_REFRESH | ACTIVATE | WINDOWDRAG |
  WINDOWDEPTH | WINDOWCLOSE, /* Flags */
  &volgadget,                   /* First gadget */
  NULL,                         /* Check mark */
  (UBYTE *)"Keyclick",  /* Title */
  NULL,                         /* Screen */
  NULL,                         /* Bitmap */
  0, 0, 0, 0,                   /* Min & Max Width & Height */
  WBENCHSCREEN                  /* Workbench window */
};

UBYTE allocation[] = {1, 8, 2, 4}; /* Channel allocation */
BYTE *waveform;     /* Pointer to waveform RAM */
struct infostruct
{
  struct Task *taskp;
  ULONG key_signal;
} ih_info;
int clup;
int volume = 63;  /* Click volume */
void handler_interface();


main()
{
  init_gadget();
  open_window();
  init_sound();
  init_info();
  open_input_device();
  add_handler();
  watch();
}
```

```
init_gadget()
{
  /* Allocate RAM for gadget's image */
  if((image = (struct Image *)
  AllocMem((long)sizeof(struct Image),
  MEMF_CHIP | MEMF_CLEAR)) == NULL)
    error("Can't allocate RAM for gadget image.");

  /* Initialize pointer to image RAM */
  volgadget.GadgetRender = (APTR)image;
  /* Set slider to horizontally-moving auto-knob */
  prop.Flags = AUTOKNOB | FREEHORIZ;
  prop.HorizPot = 0xFFFF; /* Pegged to begin with */
  prop.HorizBody = 0xFFFF / 32;
}


open_window()
{
  short exists;

  if((IntuitionBase =
  OpenLibrary("intuition.library", 1L)) == NULL)
    error("Can't open Intuition library.");

  /* Look for an already existing message port named
  "keyclick.audio."  If it exists, another key-clicker
  is active in the system and this program will abort. */
  if((ULONG)FindPort("keyclick.audio") != NULL)
  {
    /* Move window over a tad to avoid obscuring the
    original (should it remain unmoved) */
    windowdef.LeftEdge += 5;
    windowdef.TopEdge += 5;
    exists = TRUE;
  }
  else exists = FALSE;
  if((window = OpenWindow((APTR)&windowdef)) == NULL)
    error("Can't open a window.");
  if(exists == TRUE)
    error("A key-clicker is already active!");
}


init_sound()
{
  if((ioaudio = AllocMem((long)sizeof(struct IOAudio),
  MEMF_PUBLIC | MEMF_CLEAR)) == NULL)
    error("Can't allocate RAM for IOAudio structure.\n");
  if((audio_port =
  CreatePort("keyclick.audio", NULL)) == NULL)
    cleanup("Can't create audio message port.");
  ioaudio->ioa_Request.io_Message.mn_Node.ln_Pri =
    AUDIO_PRIORITY;
  ioaudio->ioa_Request.io_Message.mn_ReplyPort =
    audio_port;
  ioaudio->ioa_Data = allocation;
  ioaudio->ioa_Length = (long)sizeof(allocation);

  if((OpenDevice("audio.device", NULL, ioaudio, NULL))
  != NULL)
    error("Audio device failed to open.");
  clup |= CLUP_AUDDEV;

  if((waveform = AllocMem(WAVELENGTH, MEMF_CHIP)) == NULL)
    error("Can't allocate memory for waveform.");
  waveform[0] = 127;  waveform[1] = -127; /* Square wave */
}


init_info() /* Initializes the info structure */
{
  ih_info.taskp = FindTask(NULL); /* Pointer to TCB */
  SetTaskPri(ih_info.taskp, TASK_PRIORITY);
  ih_info.key_signal = AllocSignal(-1L);
  if(ih_info.key_signal == -1)
    error("Can't allocate a signal");
}


open_input_device()
{
  if((input_port =
  CreatePort("keyclick.input", NULL)) == NULL)
    error("Can't create an input device message port.");
  if((iostdreq = CreateStdIO(input_port)) == NULL)
    error("Can't create an IOStdReq structure.");
  if((OpenDevice("input.device", NULL, iostdreq, NULL))
  != NULL)
    error("Cannot open input device.");
  clup |= CLUP_INPDEV;
}


add_handler()
{
  interrupt.is_Code = handler_interface;
  interrupt.is_Data = (APTR)&ih_info;
  interrupt.is_Node.ln_Pri = HANDLER_PRIORITY;
  iostdreq->io_Command = IND_ADDHANDLER;
  iostdreq->io_Data = (APTR)&interrupt;
  DoIO(iostdreq);
  clup |= CLUP_HANDLER;
}


watch() /* Watches input device port for messages */
{
  struct IntuiMessage *msg, *dqmsg;
  ULONG mclass, sig;

  while((dqmsg = (struct IntuiMessage *)
  GetMsg(window->UserPort)))
    ReplyMsg(dqmsg);
  SetWindowTitles(window, -1L,
  "Keyclick V1.0 by Mike M. Duppong");
  while(1)
  {
    sig = Wait((ULONG)(1L << ih_info.key_signal) |
      (ULONG)(1L << (ULONG)window->UserPort->mp_SigBit));
    if(sig & (ULONG)(1L << ih_info.key_signal))
      click();
    else
    {
      msg = (struct IntuiMessage *)
      GetMsg(window->UserPort);
      mclass = msg->Class;
      ReplyMsg(msg); /* Reply to sender */
      switch(mclass)
      {
        case CLOSEWINDOW: cleanup(); break;
        case GADGETUP: adjust_volume(); break;
      }
    }
  }
}


adjust_volume()
{
  volume = ((long)(prop.HorizPot) * 64L) >> 16;
}


click()
{
  ioaudio->ioa_Request.io_Command = CMD_WRITE;
  ioaudio->ioa_Request.io_Flags = ADIOF_PERVOL;
  ioaudio->ioa_Data = (UBYTE *)waveform;
  ioaudio->ioa_Length = WAVELENGTH;
  ioaudio->ioa_Period = PERIOD;
  ioaudio->ioa_Volume = (ULONG)volume;
  ioaudio->ioa_Cycles = CYCLES;

  BeginIO(ioaudio); /* Send off audio command */
  if(CheckIO(ioaudio))
    error("Can't communicate with audio device.");
}


remove_handler()
{
  iostdreq->io_Command = IND_REMHANDLER;
  iostdreq->io_Data = (APTR)&interrupt;
  DoIO(iostdreq);
}
```

D - Five
Associates

19 Crosby Drive
Bedford, MA
01730-0523

(617) 275-8892

Tired of the high cost of computer repairs?

→ FLAT Labor charges
→ FREE Estimates
→ Warranty work

Also:
1764 to 512K: $61.95
128 64K vdc RAM: $40.00
NEW: C=1902 conversion to RGB-I: $40.00

$19.95*
C-64/128 & peripherals

29.95*
Amiga 1000

$34.95*
Commodore PC-10

* plus parts and sales tax

Authorized
Commodore
Service
Center

Would you use your Amiga like this?

Without ENCORE you might as well be!

Encore allows you to build macros by recording mouse movements and keystrokes. You can assign each macro to a key combination so that with a touch of the finger any macro can instantly be played back.

Macros can have spoken narration.

Macros can have scrolling text at the bottom of the monitor.

Macros can loop back for repeated playback.

Macros can be either relative or absolute.

Macros can be appended.

Macros can be user interactive.

Macros can be time or file compressed.

Macros can be of unlimited size.

Automatic mouse and keyboard lockout.

Works with any Amiga program.

Speech preferences program with phonemized word database.

No matter what you are using your Amiga for, Encore is a must for your software library. See your local dealer for details.

ENCORE is NOT COPY PROTECTED Suggested retail $69.95
Dealer inquiries invited.

ELIPSYS PO Box 16757
Seattle, Wa 98116
(206)938-0825
FAX (206)932-9520

```
struct InputEvent *input_handler(event, info_str)
register struct InputEvent *event;
register struct infostruct *info_str;
{
  if((event->ie_Class == IECLASS_RAWKEY &&
  !(event->ie_Code & IECODE_UP_PREFIX)) ||
  (event->ie_Class == IECLASS_RAWMOUSE &&
  (event->ie_Code == IECODE_LBUTTON ||
  event->ie_Code == IECODE_RBUTTON ||
  event->ie_Code == IECODE_MBUTTON)))
    Signal(info_str->taskp,
      (ULONG)(1L << info_str->key_signal));
  return(event);
}


error(message)
char *message;
{
  struct IntuiMessage *msg;
  ULONG mclass = NULL;

  /* If the window opened okay, place the error message in
  the Workbench screen's title bar and wait for the window
  to close.  Otherwise, attempt to output the message to
  a CLI window with puts(). */
  if(window != NULL) /* If opened successfully... */
  {
    SetWindowTitles(window, -1L, message);
    while(mclass != CLOSEWINDOW)
    {
      WaitPort(window->UserPort); /* Wait for a message */
      msg = (struct IntuiMessage *)
       GetMsg(window->UserPort);
      mclass = msg->Class; /* Grab message class */
      ReplyMsg(msg); /* Reply to message */
    }
  }
  else puts(message); /* Try to output to CLI. */
  cleanup(); /* Do the standard cleanup from here. */
}


cleanup()
{
  struct Message *dqmsg;

  if(clup & CLUP_HANDLER)
    remove_handler();
  if(ih_info.key_signal != -1)
    FreeSignal((ULONG)(1L << ih_info.key_signal));
  if(clup & CLUP_INPDEV)
    CloseDevice(iostdreq);
  if(iostdreq)
    DeleteStdIO(iostdreq);
  if(clup & CLUP_AUDDEV)
    CloseDevice(ioaudio);
  if(waveform != NULL)
    FreeMem(waveform, WAVELENGTH);
  if(ioaudio != NULL)
    FreeMem(ioaudio, (long)sizeof(struct IOAudio));
  if(input_port)
    DeletePort(input_port);
  if(audio_port)
    DeletePort(audio_port);
  if(window != NULL)
  { /* Make sure all messages are answered */
    while((dqmsg = GetMsg(window->UserPort)))
      ReplyMsg(dqmsg);
    CloseWindow(window);
  }
  if(image != NULL)
    FreeMem(image, (long)sizeof(struct Image));
  if(IntuitionBase != NULL)
    CloseLibrary(IntuitionBase);
  exit(NULL);
}


#asm
  cseg
  xref _input_handler
  xdef _handler_interface

_handler_interface:
  move.l a4,-(a7)
  movem.l a0-a1,-(a7)
  jsr _geta4#
  jsr _input_handler
  addq.l #8,a7
  move.l (a7)+,a4
  rts
#endasm
```

•AC•

MORE ABOUT LINKED LISTS IN C

Techniques and Applications

by Forest W. Arnold

In AC V3.7, William Gammill detailed linked lists and their usefulness for managing dynamic memory. This article presents two clever macros for dealing with doubly-linked lists. In this article, I discuss singly-linked lists and list algorithms, develop procedures for managing the lists, show how different data types can be stored in the same list, and finally, present a simple graphics program to demonstrate how lists can be used.

Why Use Lists?

The first time I used a programming language with facilities for building linked lists, I couldn't see how anything which looked so complicated could be worth the effort. At the simplest level—a list storing only one kind of data—the only advantage I could see for using lists (instead of arrays or matrices) was that I didn't need to know in advance how much memory the data needed. This dynamic aspect of lists is clearly an advantage over matrices, but I still didn't realize how powerful linked lists were. That is, until I learned how to store different data in the same list and how to build tree structures.

When combined with C structures, function pointers, and recursion, linked lists can implement very complex program and data structures with surprisingly simple and concise techniques. One example is the Amiga operating system and the Intuition software built on top of the system software. All Amiga and Intuition structures, tasks, files, and devices are created, managed, and destroyed using lists and queues. Lists and queues are implemented and manipulated with a only a few procedures. Much of the Amiga's power, flexibility, and elegance can be attributed to the dynamic nature and generality of lists and queues.

Another example of the power of linked lists is provided by LISP, which combines both code and data into lists, lists of lists, lists of lists of lists, ad nauseam. It accomplishes this magic through these list structures and a single recursive "eval" routine (which "evaluates" the lists).

At first glance, structures and programs built from lists and trees appear complex, but the techniques and algorithms to construct and manage complex trees of program structures and data structures are the same as those used for a single linked list. Once you understand how to work with one linked list, you understand them all, no matter how elaborate. The following section describes two variations of the simplest type of list—the singly-linked list. Doubly-linked lists are also briefly described.

Basic Linked List Structure

If you have read some popular C programming books or the linked lists article in AC V3.7, you probably already have some idea what a linked list is. If you are just starting with lists, though, this section should help you better understand the basic structure of linked lists.

A linked list is a sequence of objects (nodes) connected by pointers. Each list has a start node (the list head or top) and an end node (the list tail). Each node in the list has the same structure, and contains a link which specifies how to access (at a minimum) the next element in the list. A simple node for a singly-linked list can be defined in C as follows:

```
typedef struct node
{
  struct node     *next;  /* pointer to next node  */
  unsigned char   *data;  /* pointer to the data   */
} NODE_T;
```

The pointer to the next node in the list ("next") is called the link, since it strings the nodes together. Using type "unsigned char *" to point to the node's data lets you see the node structure as a "hook." The "hook" points to any type of data structure you need and maintains a consistent set of procedures for manipulating the list structure, no matter how complex things get.

When the data pointer is placed in the node, the type is cast to "unsigned char *"; when the data is accessed, the type is re-cast to the correct type. The cast operator ensures that the compiler correctly generates offsets into the pointed-to structure, and prevents the compiler from generating a lot of messy warning messages.

A doubly-linked list can be created by adding another pointer to the above structure. The doubly-linked list node is defined as follows:

```
typedef struct node
{
  struct node   *prev;  /* pointer to previous node  */
  struct node   *next;  /* pointer to next node      */
  unsigned char *data;  /* pointer to node's data    */
} DNODE_T;
```

A singly-linked list can only be traversed from a given start node in the list to the end node. The start node can be any node in the list that can be accessed. A doubly-linked list can be

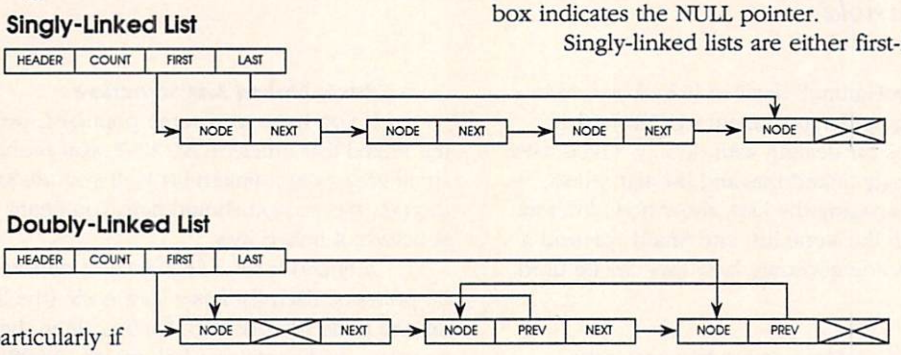
(continued)

traversed from any node in the list—either back to the beginning or forward to the end. Exec's lists are doubly-linked lists, while Intuition's menu structures are singly-linked lists.

Memory for both the list nodes and the pointed-to data structures is usually dynamically allocated as needed, and freed when no longer needed. This way the data structures can grow as large as memory permits, while using only the memory actually required. Dynamic memory allocation and deallocation are major benefits of linked lists. Statically allocated structures, such as arrays, always seem bigger than anyone ever needs. (Except the first real user.)

To "anchor" the list to a place in memory where it can be easily accessed, a list header structure is generally used (but not strictly necessary). The header is another C structure which contains at least a pointer to the first node in the list. It may also contain a pointer to the last node in the list, particularly if the list is doubly-linked or implements a queue. Other information frequently stored in the header includes a tally of nodes, a name for the list, and information identifying the type of data the nodes point to if all data in the list are the same type. A typical list header could be defined in C as follows:

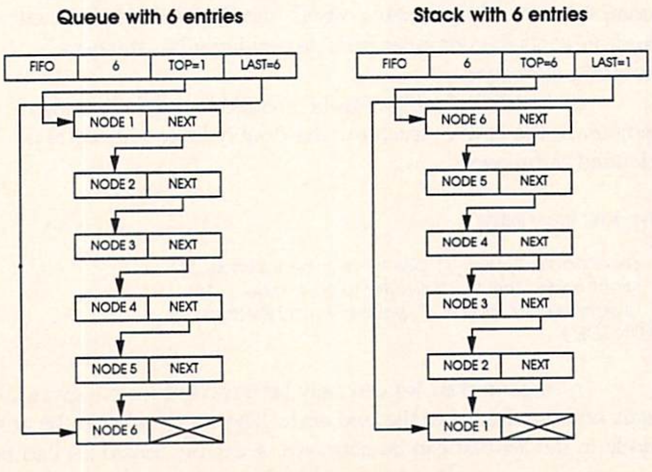
Figure One



```
typedef struct listHeader
{
        struct node *top;          /* first node in the list  */
        struct node *tail;         /* last node in the list  */
        int dType;                 /* data type for the nodes */
        int count;                 /* number of nodes        */
} HEADER_T;
```

The list header structure Exec uses is almost identical to this example.

Figure Two


Queue with 6 entries

Stack with 6 entries

Queues and Stacks

If different types of data are stored in the same list, the 'dType' field is stored in the node structure instead of in the list header (since each node can carry around different data). The demonstration program in Listing 1 uses this technique to store both "point" structures and "line" structures in one singly-linked list. Another variation of the node structure for storing different data in the same list uses C's union construct to hold the actual data or a pointer to the data (rather than an untyped pointer to the data). The disadvantage of storing the data using a union is that space is wasted if the data types are different lengths.

Figure 1 depicts a singly-linked list and a doubly-linked list. When drawing linked lists, the nodes are usually shown as rectangular boxes and the links (pointers) are shown as arrows pointing to the "next" (and "previous") node. An "X" in a "next" box indicates the NULL pointer.

Singly-linked lists are either first-in, first-out (FIFO) lists, or last-in, first-out (LIFO) lists. In FIFO lists, nodes are always removed from the "top" of the list, and the first node placed in the list is the first node removed from the list. In LIFO lists, the last node placed in the list is the first node removed. When queues and stacks are implemented as lists, FIFO lists are used for queues and LIFO lists are used for stacks because of the insertion-removal relationship. Figure 2 shows list representations of both queues and stacks; the demonstration program creates both types of lists and prints them to files.

The basic data structures for all linked lists have now been presented, and all that is left to describe are the actual operations needed to manage lists.

List Operations

A nice thing about linked lists is that there are only a few basic operations for lists; there are many variations in implementation, but the fundamental algorithms are the same. The basic operations are: (1) create a list; (2) create a node and its data; (3) insert a node in the list; (4) search a list for a node; (5) remove a node from the list; (6) delete a node and its data; and (7) delete a list.

Once mastered, these operations can be applied to any linked list. In fact, if data type dependencies are kept out of the implementation, the exact same code can be used for any list. (This was the purpose of the macros described in AC V3.7). This portability suggests packaging the node structure definitions and list operations into a single file as an abstract data type, usable without modification by any application. The code in the demonstration program does contain data dependencies. In my next article, I'll discuss how function pointers can be used to eliminate type dependencies.

In the following paragraphs, the basic operations for singly-linked lists are described as sequences of steps. Each description is followed by a description of the actual procedure in the demonstration program that implements the operation. The operations are meant for lists that can store pointers to different data types in each node. Aside from removing nodes from a list, all the operations are very simple and can be implemented with only a few lines of code.

List creation:

1. Allocate space for the list header.
2. Set the list "top" and "tail" pointers to NULL; set the optional "count" field to zero.

In the demo program, the list headers are declared as actual structure instances (instead of being dynamically allocated) because only two are used. The header structure is "typedef'ed" as HEADER_T. The header is an "empty" list into which nodes can be inserted.

Node-data creation:

1. Allocate space for the node.
2. Allocate space for the pointed-to data.
3. Initialize the variables defining the data to the appropriate values.
4. Set the node data-type field to the code that corresponds to the data structure.
5. Set the node "data" pointer to the address of the initialized data structure, type-cast as 'unsigned char *'
6. Set the node "next" field to NULL.

The demo program uses two data types: a POINT_TYPE and a LINE_TYPE. These correspond to the POINT_T data structure and the LINE_T data structure and are #define'd as integer values. The procedures createPoint() and createLine() are implementations of node-data creation and initialization. Both procedures return pointers to the newly-created and initialized NODE_T data structure.

Next, the node must be inserted into the node list. The method used depends on whether the list is a FIFO list or a LIFO list. For FIFO lists, new nodes are always added to the end of the list. For LIFO lists, new nodes are always added to the beginning of the list.

LIFO list node insertion:

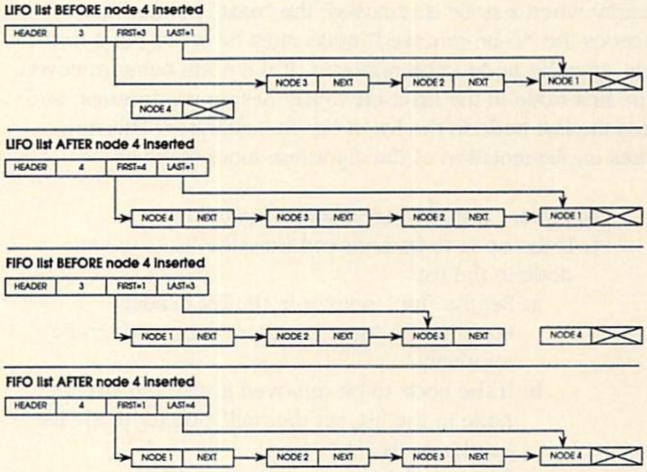
1. Set the new node's "next" pointer equal to the "top" pointer in the list header.
2. Set the "top" pointer in the list header equal to the address of the newly-added node.
3. If the node is the first added to the list, set the "tail" pointer in the list header equal to the address of the new node.
4. (Optional) Increment the list node counter variable.

FIFO list node insertion:

1. If the new node is the first node added to the list:
    a. Set the "top" pointer in the list header equal to the address of the new node.
    b. Set the "last" pointer in the list header equal to the address of the new node.
2. Otherwise, set the "next" pointer of the last node in the list equal to the address of the new node.
3. Set the "tail" pointer in the list header equal to the new node's address .
4. (Optional) Increment the list node counter variable.

The procedures insertInStack() and insertInQueue() in the demo program are implementations of list node insertion. Both procedures take a list header's address and the address of a node to add to the list as input arguments and then add the node to the list. Figure 3 shows "before" and "after" pictures of list node insertion.

Figure Three



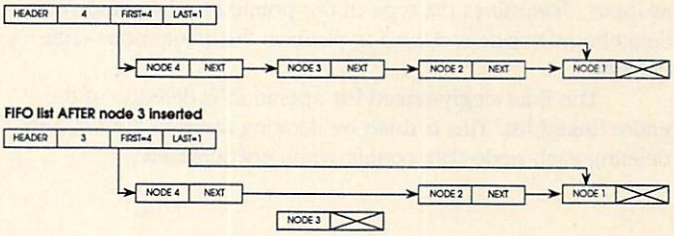
Linked List Node Insertion

To search a list for a node:

1. Set the current node equal to the "top" node in the list header.
2. If the current node is NULL, the end of the list has been found. Return NULL.
3. Otherwise, compare the node or the node's data to the search criteria.
4. If a match is found, return the pointer to the node.
5. Otherwise, set the current node equal to the address of the node stored in the current node's "next" pointer.
6. Go to step 2.

Searching a singly-linked list follows the chain of link ("next") pointers. The process is often called "pointer chasing." The actual search criteria used depends upon the application. In the demo program, the search criteria are (x,y) coordinates generated by a mouse select button event. SearchList() implements the coordinate search and returns either a pointer to the found node or a NULL pointer.

Figure Four



Linked List Node Removal

(continued)

The operation to remove a node from a list is more complicated than the other list operations. To preserve the list's integrity when a node is removed, the "next" pointer that precedes the "to-be-removed" node must be set to point to the node *after* the node to be removed. If the node being removed is the first node in the list it obviously has no predecessor, and if it is the last node in the list, it has no successor. This quirk makes implementation of the algorithm more involved.

To remove a node from a singly-linked list:

1. If the node to be removed from the list is the top node in the list:
    a. Set the "top" pointer in the list header equal to the "next" pointer of the node to be removed.
    b. If the node to be removed is also the last node in the list, set the "tail" pointer in the list header to "NULL."
    c. (Optional) Decrement the list node counter.
    d. Return "success."
2. Otherwise, set the previous node equal to top node.
3. Set the current node equal to the previous node's "next" pointer.
4. If the current node is "NULL," return "failure."
5. If the node is equal to the node to be removed:
    a. Set the previous node's "next" pointer equal to the current node's "next" pointer.
    b. If the current node is also the last node in the list, set the "tail" pointer in the list header to point to the previous node.
    c. (Optional) Decrement the list node counter.
    d. Return "success."
6. Otherwise, set the previous node equal to the current node. Go to step 3.

If the list is a LIFO list, the "tail" pointer in the list header is not needed, and node removal can be simplified by eliminating the steps that check for removal of the last node. The procedure removeNode() in the demo program implements the algorithm shown above. Figure 4 is a diagram of a list before and after removal of a node.

To delete a node-data couple:

1. If the node pointer is NULL, return "failure."
2. Otherwise, if the node data pointer is not NULL, free( ) the "pointed-to" data structure.
3. Free( ) the node structure. Return "success."

The demo procedure deleteNode( ) takes a node pointer as input, determines the type of the pointed-to data, and calls deletePoint() or deleteLine() to perform the actual node-data deletion.

The final singly-linked list operation is deletion of the entire linked list. This is done by iterating through the list, and deleting each node-data couple when encountered.

To delete an entire singly-linked list:

1. Set the current node equal to the list "top" node.
2. If the current node is NULL:
    a. set the "last" node pointer in the list header to NULL.
    b. return "success."
3. Otherwise, set the list "top" node pointer equal to the current node's "next" pointer.
4. Delete the current node.
5. (Optional) Decrement the list node counter.
6. Go to step 1.

We've now seen all the basic list operations. Before we move on to another topic, I'll make a few comments. First, unless adding the node-data couples to the end of a list is necessary (for a queue, for example), the tail pointer to the last node in the list is unnecessary. Second, in many cases, the list header itself can be eliminated. Unless information that applies to the list as a whole is needed, the list can be accessed by a statically-allocated node as the first node in the list.

Finally, in cases that require many small memory allocations and deallocations, using "malloc" (or "calloc") and "free" can be time-consuming. If speed is a primary consideration, implement an internal memory management scheme, allocate a large chunk of memory at start-up time, and use custom written procedures to allocate and deallocate small segments of memory.

Disadvantages of Using Lists

Aside from memory allocation speed, the only drawback I find with linked lists is that to access any given node, the list must be searched sequentially. This is true for both doubly- and singly-linked lists. Even when data is inserted to the list in sorted order, the list must still be sequentially searched. Several techniques (beyond the scope of this article) can overcome the search drawback. Such techniques allow a node to be found in the time it takes to perform a binary search of an array.

Putting Lists to Work

Seeing uses for linked lists is much more informative than reading a description of how they work. So, in addition to the procedures that implement the basic list operations, the demonstration program also contains the following procedures to manipulate lists:

1. Procedure reportList() prints the node structure and the pointed-to data structure to a file.
2. Procedure displayList() graphically displays the pointed-to data (points and lines) in an Intuition window.
3. Procedure displayNodeInfo() highlights a picked point or line and displays the point or line data in a text message window. This procedure calls display PointInfo() or displayLineInfo() to highlight and display the data.

LISTDEMO.C—The Demonstration Program

The demo program in Listing 1 is contained in a single file and can be compiled and linked in one step. The Lattice compile and link command is 'lc -L listDemo.c'. The program is run from a CLI window.

The program creates both a FIFO list and a LIFO list, each containing five point data types and five line data types. The program prints each list to a separate files in the current directory, so you can see the difference between the lists. The FIFO list output file is "queue.out," and the LIFO list output file is "stack.out". The FIFO list is then deleted since it is not used for the remainder of the program. The program also deletes the first and last entries in the LIFO list, and again prints the list after each node-data couple is deleted.

Then the program gets a little more interesting. Intuition is set up, a window is opened, and the LIFO list data is graphically displayed. The program calls handleInput( ), an input handling procedure that monitors the IDCMP until the mouse select button is pressed and released. The mouse cursor coordinates are sent to the procedure searchList(), which looks for a node at the selected coordinates.

If the selected coordinates are within four pixels of a point (displayed as an "x") or line segment, searchList( ) returns a pointer to the node that points to the point or line. The input handler then halts monitoring the IDCMP and calls displayNodeInfo( ) with a pointer to the found node. The point or line is "highlighted," and gpDisplayMsg( ) displays the data for both the node and the picked point or line in a "message" window. Closing the message window with the close gadget returns control to handleInput(), which then resumes monitoring input events. Closing the graphical display window with the close gadget stops the program.

I mentioned earlier that some list procedures contain data type dependencies. These procedures are reportList( ), searchList( ), displayList( ), displayNodeInfo( ), and deleteNode( ). Each procedure makes specific comparisons of the dataType variable stored in the node to determine if the "pointed-to" data defines a point or a line. The procedure then calls the appropriate point or line procedure.

Embedding type dependencies in a procedure guarantees that the procedure can be used only for the specific types coded into the procedure. In my next article, I'll discuss a technique that uses function pointers and function pointer tables to remove type dependencies from the code. I'll use this same demonstration program (with modifications) to demonstrate the function pointers.

Conclusion

Lists containing only primitive node-data couples are handy and provide a consistent mechanism for manipulating varied types of data. The real power of lists, though, is not evident until they are used in tree structures. Since a node in a list can point to anything (including procedures), the node can also point to another list. By constructing lists whose elements are also lists, very complex, hierarchical data structures can be easily implemented and managed with only the simple methods discussed above. In fact, if properly constructed and combined with the proper control mechanisms, an entire database can be managed by simply managing the single record that contains the top-level header or node.

Once you understand pointers and master the algorithms presented in this article, you should be able to begin exploiting the full power of linked lists in C.

Further Reading

Two of the best, most understandable books covering linked lists I have found are:

Advanced C: Techniques & Applications, Gerald E. Sobelman and David E. Krekelberg, Que Corporation.

Data Structures and C Programs, Christopher J. Van Wyk, Addison-Wesley.

Advanced C: Techniques & Applications, is available in paperback and is relatively inexpensive. If you are interested in applying lists to graphics programs, I highly recommend it. *Data Structures and C Programs* is more theoretical and covers a wide range of advanced topics (applied theory with many good examples in C). I recommend it for anyone who has passed the intermediate C programming stage.

Until next time - keep programming!

Listing One

```
/*=======================================================*/
/*  Listing 1.  Linked List Demo Program Source Code
 *
 *  listDemo.c - program to demonstrate generic linked
 *               lists with different data types
 *
 *  (c)  Copyright 1988 by Forest W. Arnold
*/
/*=======================================================*/

#include <exec/types.h>
#include <intuition/intuition.h>
#include <stdio.h>

#define INTUITION_REV 33
#define GRAPHICS_REV  33

struct IntuitionBase *IntuitionBase;
struct GfxBase       *GfxBase;

/*  'make life easy' stuff                                */

#define BOOLEAN   int
#ifndef FALSE
#define FALSE     0
#endif
#ifndef TRUE
#define TRUE      1
#endif
#ifndef NOT
#define NOT       !
#endif

#define NEW(x)    ( (x *)malloc( sizeof(x) ) )
#define FREE(x)   ( free( (char *)x ) )
#define ABS(x)    ( (x) < 0 ? -(x) : (x) )
#define MIN(x,y)  ( (x) < (y) ? (x) : (y) )
#define MAX(x,y)  ( (x) > (y) ? (x) : (y) )

/*
 *  define some data types to put into the list
 *  We'll use a simple pixel point and define a line
 *  using two instances of a point.
*/

typedef struct point
{
   short    x;
   short    y;
} POINT_T;

#define POINT_TYPE   1

typedef struct line
{
   short  x1,y1;
   short  x2,y2;
} LINE_T;
```

```
#define LINE_TYPE    2

/*
 *  define a node for the linked list.  Note that we use
 *  a pointer to ‘unsigned char’ to point to the actual
 *  data so we can point to anything.
*/

typedef struct node
{
   struct node   *next;    /* the list ‘link’ pointer    */
   short         dataType; /* code for type of data      */
   unsigned char *data;    /* pointer to the actual data */
} NODE_T;

/*
 *  define a generic list header.  For ‘stacks’, only the
 *  topNode ptr is needed.  For ‘queues’, both node
 *  pointers are needed.  The count variable is optional
*/

typedef struct header
{
   NODE_T   *topNode;
   NODE_T   *lastNode;
   int      count;
} HEADER_T;

/* define the procedures for manipulating the data types */

NODE_T  *createPoint();    /* return point & node        */
BOOLEAN searchPoint();
BOOLEAN deletePoint();
BOOLEAN displayPoint();
BOOLEAN displayPointInfo();
BOOLEAN reportPoint();

NODE_T  *createLine();     /* return line & node         */
BOOLEAN searchLine();
BOOLEAN deleteLine();
BOOLEAN displayLine();
BOOLEAN displayLineInfo();
BOOLEAN reportLine();

BOOLEAN insertInStack();   /* ‘push’ node onto node list */
BOOLEAN insertInQueue();   /* append node to end of list */
BOOLEAN removeNode();      /* remove node from list      */
BOOLEAN deleteNode();      /* delete a node & data       */
BOOLEAN deleteList();      /* delete entire list         */
BOOLEAN displayList();     /* display entire list        */
BOOLEAN displayListInfo(); /* display info about a node  */
BOOLEAN reportList();      /* output info about list     */
NODE_T  *searchList();     /* search list for a datatype */

/*  intuition interface routines                         */

BOOLEAN        openLibs();
void           closeLibs();
void           cleanUp();
struct Window *allocWindow();
void           gpDisplayMsg();
void           handleInput();

/*------------------------------*/

main()
{
   struct Window  *window;
   HEADER_T       stackHead,queueHead;
   NODE_T         *sNode,*qNode;
   FILE           *stackFile,*queueFile;
   unsigned long  wFlags;
   unsigned short iFlags;
   short          i,x,y,x1,y1;
   short          xInc,yInc,temp;

   wFlags = WINDOWCLOSE|WINDOWDRAG|SMART_REFRESH|ACTIVATE;
   iFlags = CLOSEWINDOW|MOUSEBUTTONS;

   /*
    *  initialize the list headers.  Make sure pointers
    *  start out as ‘NULL’. This will be the signal that
    *  the end of the list has been reached
   */

   stackHead.topNode = stackHead.lastNode = NULL;
   queueHead.topNode = queueHead.lastNode = NULL;
   stackHead.count = queueHead.count = 0;

   /*
    *  create the lists, alternating between points and
    *  lines.
   */
   x = 120; y = 30;
   x1 = 135; y1 = 20;
   xInc = 40; yInc = 0;

   for ( i = 0; i < 5; i++ )
   {
      if ( sNode = createPoint( x,y ) )
         (void) insertInStack( &stackHead,sNode );
      if ( qNode = createPoint( x,y ) )
         (void) insertInQueue( &queueHead,qNode );
      if ( sNode = createLine( x1,y1,x1+xInc,y1+yInc ) )
         (void) insertInStack( &stackHead,sNode );
      if ( qNode = createLine( x1,y1,x1+xInc,y1+yInc ) )
         (void) insertInQueue( &queueHead,qNode );

      x += 60; y += 20;
      x1 += 60; y1 += 20;
      temp = xInc;  xInc = yInc;  yInc = temp;
   }

   /*  output both lists to a file                         */

   if ( NOT (stackFile = fopen(“stack.out”,”w”)) ||
        NOT (queueFile = fopen(“queue.out”,”w”))   )
   {
      printf(“Can’t open output files.”);
      (void) deleteList( &queueHead );
      (void) deleteList( &stackHead );
      exit(1);
   }
   fprintf(stackFile,”Initial stack:\n\n”);
   fprintf(queueFile,”Initial queue:\n\n”);
   (void) reportList( stackFile,&stackHead );
   (void) reportList( queueFile,&queueHead );

   /*  free the queue since we’ll nc longer use it         */

   (void) deleteList( &queueHead );
   fflush(queueFile);  fclose(queueFile);

   /*  search for & remove first & last node               */

   if ( sNode = searchList(&stackHead,375,100) )
   {
      (void) removeNode(&stackHead,sNode);
      (void) deleteNode(sNode);
      fprintf(stackFile,”\n\nStack after deleting “);
      fprintf(stackFile,”top node:\n\n”);
      (void) reportList( stackFile,&stackHead );
   }
   if ( sNode = searchList(&stackHead,120,30) )
   {
      (void) removeNode(&stackHead,sNode);
      (void) deleteNode(sNode);
      fprintf(stackFile,”\n\nStack after deleting “);
      fprintf(stackFile,”last node:\n\n”);
      (void) reportList( stackFile,&stackHead );
   }
   fflush(stackFile);  fclose(stackFile);

   /* initialize intuition stuff & display the data        */

   if ( NOT openLibs() )
   {
      cleanUp(&stackHead,NULL);
      exit(1);
   }
   if ( NOT (window=allocWindow(NULL,NULL,40,20,560,150,
                       “LIST DEMO”,wFlags,iFlags,-1,-1)) )
   {
      cleanUp(&stackHead,NULL);
      exit(1);
   }
   SetAPen(window->RPort,1);
   Move(window->RPort,180,140);
   Text(window->RPort,”Pick entity to show data.”,25);
   (void)displayList( window,&stackHead );

   /* process input until window close gadget picked      */

   handleInput(window,&stackHead);
   cleanUp( &stackHead,window );
   exit(0);
}
/*------------------------------*/
void cleanUp(hdr,window)
HEADER_T      *hdr;
struct Window *window;
{
   if ( hdr ) deleteList( hdr );
   if ( window ) CloseWindow( window );
   closeLibs();
}
/*===================================================*/
/*
 *  insertInStack() - insert a node into a stack by
 *                    ‘pushing’ it onto top of list
 *
 *  for a ‘stack’, we simply make the node’s ‘next’ ptr
 *  point to the top node in the list, then make the
 *  node the top of the list
```

```
*/
/*=======================================================*/

BOOLEAN insertInStack(hdr,node)
HEADER_T   *hdr;
NODE_T     *node;
{
   /*  check for valid hdr & node pointers              */

   if ( NOT hdr || NOT node ) return( FALSE );

   node->next = hdr->topNode; /* link it                */
   hdr->topNode = node;       /* make node the new 'top' */

   /*
    *  update ptr to last node if this is first node in
    *  the list, then update node count
   */
   if ( NOT hdr->lastNode )
      hdr->lastNode = node;

   hdr->count++;
   return( TRUE );
}
/*=======================================================*/
/*
 *  insertInQueue() - insert a node into a queue by
 *                    appending it to bottom of list.
 *
 *  for a ‘queue’, we simply make the last node in the
 *  list point to the new node, make the new node point to
 *  NULL, and update the headers ‘last node’ pointer.
*/
/*=======================================================*/

BOOLEAN insertInQueue(hdr,node)
HEADER_T   *hdr;
NODE_T     *node;
{
   /*  check for valid hdr & node pointers              */

   if ( NOT hdr || NOT node ) return( FALSE );

   node->next = NULL;

   if ( hdr->lastNode )              /* is list empty?    */
   {                                 /* Not empty         */
      hdr->lastNode->next = node;    /* Append node       */
      hdr->lastNode = node;          /* update header     */
   }
   else                              /* list is empty     */
   {
      hdr->topNode = node;           /* update both top   */
      hdr->lastNode = node;          /* and last pointers */
   }
   hdr->count++;
   return( TRUE );
}
/*=======================================================*/
/*
 *  BOOLEAN deleteList() - delete an entire linked list
 *
 *  follow list pointers from top to bottom, freeing the
 *  data, then the node.  Update top pointer as we go.
*/
/*=======================================================*/

BOOLEAN deleteList( hdr )
HEADER_T *hdr;
{
   NODE_T   *node;   /*  the current node to delete      */

   if ( NOT hdr ) return ( FALSE );

   while ( node = hdr->topNode ) /* stop when ‘NULL’      */
   {
      hdr->topNode = node->next; /* update list top ptr  */
      (void) deleteNode( node ); /* delete it            */
      hdr->count—;
   }
   hdr->lastNode = NULL;         /* update hdr ‘last’ ptr*/
   return( TRUE );
}
/*=======================================================*/
/*
 *  reportList() - print out entire list to a file
 *
 *  traverse list from top to bottom, calling appropriate
 *  output routine for the data type
*/
/*=======================================================*/

BOOLEAN reportList( outFile,hdr )
FILE     *outFile;
HEADER_T *hdr;
{
   NODE_T   *node;   /*  the current node                */

   if ( NOT outFile || NOT hdr ) return( FALSE );

   fprintf(outFile,”Linked list output:\n”);
   fprintf(outFile,”topNode = %x, lastNode = %x\n”,
           hdr->topNode,hdr->lastNode);
   fprintf(outFile,”Number of nodes = %d\n”,hdr->count);

   node = hdr->topNode;

   while ( node )     /*  stop when pointer is NULL       */
   {
      fprintf(outFile,”\nnode = %x, next = %x\n”,
              node,node->next);
      fprintf(outFile,”dataType = %d, data ptr = %x\n”,
              node->dataType,node->data);

      /* get type and call appropriate report routine     */

      if ( node->dataType == POINT_TYPE )
         (void) reportPoint( outFile,node );
      else
      if ( node->dataType == LINE_TYPE )
         (void) reportLine( outFile,node );

      node = node->next;   /*  get ptr to next in list    */
   }
   return( TRUE );
}
/*=======================================================*/
/*
 *  searchList() - look for a node at input coordinates
*/
/*=======================================================*/

NODE_T *searchList( hdr,x,y )
HEADER_T *hdr;
short    x,y;
{
   NODE_T   *node;          /* the current search node    */

   if ( NOT hdr ) return( FALSE );

   node = hdr->topNode;     /* start at top               */

   while( node )            /* stop when node is NULL     */
   {
      if ( node->dataType == POINT_TYPE )
      {
         if ( searchPoint(node,x,y) )
            return( node );           /*  found it        */
      }
      if ( node->dataType == LINE_TYPE )
      {
         if ( searchLine(node,x,y) )
            return( node );           /*  found it        */
      }
      node = node->next;   /* get next node in the list  */
   }
   return( NULL );          /* didn’t find one at the x,y */
}
/*=======================================================*/
/*
 *  displayList() - graphically display entire list
 *  displayNodeInfo() - text display of data for a node
*/
/*=======================================================*/

BOOLEAN displayList( window,hdr )
struct Window  *window;
HEADER_T       *hdr;
{
   NODE_T   *node;          /* the current node           */

   if ( NOT hdr || NOT window ) return( FALSE );

   node = hdr->topNode;     /* start at top               */

   while( node )            /* stop when node is NULL     */
   {
      if ( node->dataType == POINT_TYPE )
         (void) displayPoint( window,node );
      else
      if ( node->dataType == LINE_TYPE )
         (void) displayLine( window,node  );

      node = node->next;   /* get next node in the list  */
   }
   return( TRUE );
}
/*-----------------------------*/

BOOLEAN displayNodeInfo( window,node )
struct Window  *window;
NODE_T         *node;
{
   if ( NOT node || NOT window ) return( FALSE );
   if ( node->dataType == POINT_TYPE )
      (void) displayPointInfo( window,node );
   else
   if ( node->dataType == LINE_TYPE )
      (void) displayLineInfo( window,node  );

   return( TRUE );
}
/*====================================================*/
/*
 *  removeNode() - remove a node from a list.  This is
 *                 the trickiest one
*/
/*====================================================*/

BOOLEAN removeNode(hdr,node)
HEADER_T *hdr;
NODE_T   *node;
{
   NODE_T   *prev,*curr;   /* previous,current nodes      */

   if ( NOT hdr || NOT node ) return( FALSE );

   /*  check the top node first                            */

   if ( NOT (prev = hdr->topNode) )
      return( FALSE );      /* list is empty              */

   if ( prev == node )      /* is it the top node ?        */
   {
      hdr->topNode = prev->next;     /* re-link nodes     */

      if ( prev == hdr->lastNode )  /* is it also last ? */
         hdr->lastNode = NULL;

      hdr->count–;
      return( TRUE );
   }

   /* it wasn't the first one - have to search the list  */

   while( curr = prev->next )
   {
      if ( node == curr )            /* is this it ?      */
      {
         prev->next = node->next;    /* re-link nodes     */

         if(node == hdr->lastNode)   /* is it also last ? */
            hdr->lastNode = prev;

         hdr->count–;
         return( TRUE );
      }
      prev = curr;                   /* reset previous    */
   }
   return( FALSE );                  /* not in list       */
}
/*====================================================*/
/*
 *  deleteNode() - delete a single node & its data
*/
/*====================================================*/

BOOLEAN deleteNode( node )
NODE_T   *node;
{
   if ( NOT node ) return( FALSE );

   if ( node->dataType == POINT_TYPE )
      return( deletePoint( node ) );
   else
   if ( node->dataType == LINE_TYPE )
      return( deleteLine( node ) );
   else
      return( FALSE );
}
/*====================================================*/
/*
 *  Procedures for points:
 *  createPoint() - create and initialize a point & its
 *                  list node
 *  deletePoint() - delete an instance of a point
 *  reportPoint() - output info about a point instance
 *  searchPoint() - see if a point is at input coordinates
 *  displayPoint() - graphically display a point
 *  displayPointInfo() - display data in message window
*/
/*====================================================*/

NODE_T *createPoint( x,y )
short    x,y;
{
   NODE_T   *node;   /* the list node to carry the point */
   POINT_T  *point;  /* the point structure              */

   if ( NOT (node=NEW(NODE_T)) )   /* allocate the node  */
```

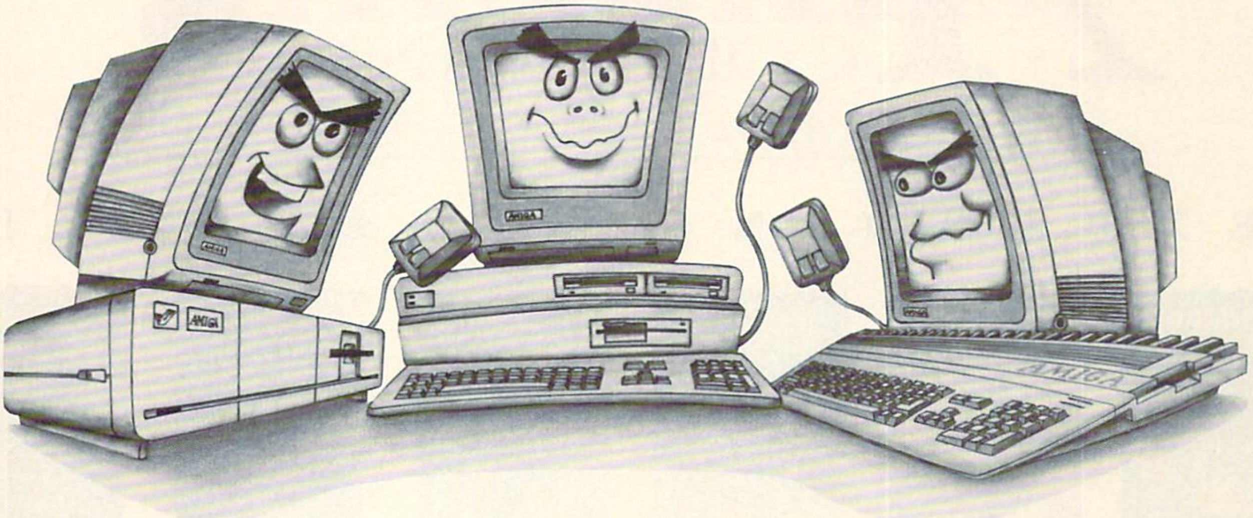
```
      return( NULL );

   if ( NOT (point=NEW(POINT_T)) ) /* allocate the point */
   {                               /* didn't get it      */
      FREE( node );                /* release node memory*/
      return( NULL );
   }
   point->x = x;  point->y = y;    /* init the point     */

   /* initialize the node. Coerce point to unsigned char */

   node->next = NULL;
   node->dataType = POINT_TYPE;
   node->data = (unsigned char*)point;
   return( node );
}
/*——————————————————————————*/
BOOLEAN deletePoint( node )
NODE_T   *node;
{
   POINT_T  *point;

   /*
    *  make sure we have a valid node pointer, correct
    *  data type, and valid pointer to free
   */

   if ( NOT node || node->dataType != POINT_TYPE )
      return( FALSE );
   if ( NOT (point = (POINT_T*)node->data) )
      return( FALSE );

   FREE( point );
   FREE( node );
   return( TRUE );
}
/*——————————————————————————*/
BOOLEAN reportPoint( outFile,node )
FILE     *outFile;
NODE_T   *node;
{
   POINT_T  *point;

   /*  make sure pointer is valid  & type is correct      */

   if ( NOT node || node->dataType != POINT_TYPE )
      return( FALSE );
   if ( NOT (point = (POINT_T*)node->data) )
      return( FALSE );

   fprintf(outFile,”>  point = %x, x = %d, y = %d\n”,
          point,point->x,point->y);

   return( TRUE );
}
/*——————————————————————————*/
BOOLEAN searchPoint( node,x,y )
NODE_T   *node;
short    x,y;
{
   POINT_T  *point;
   short    xDist,yDist;
   short    eps = 4;

   if ( NOT node || NOT (point = (POINT_T*)node->data) )
      return( FALSE );

   xDist = ABS( x - point->x );
   yDist = ABS( y - point->y );

   if ( xDist <= eps && yDist <= eps )
      return( TRUE );

   return( FALSE );
}
/*——————————————————————————*/
BOOLEAN displayPoint( window,node )
struct Window *window;
NODE_T        *node;
{
   POINT_T *point;
   short   x,y;

   /* make sure we have valid data, the display it        */

   if ( NOT window || NOT node ) return( FALSE );
   if ( NOT (point = (POINT_T *)node->data)) return(FALSE);

   x = point->x;  y = point->y;

   Move(window->RPort,x-2,y-2);
   Draw(window->RPort,x+2,y+2);
   Move(window->RPort,x-2,y+2);
   Draw(window->RPort,x+2,y-2);
   return( TRUE );
}
/*——————————————————————————*/
```

```
BOOLEAN displayPointInfo( window,node )
struct Window *window;
NODE_T        *node;
{
   POINT_T *point;
   char    str1[61],str2[61],str3[61];

   /* make sure we have valid data, then display it    */

   if ( NOT window || NOT node ) return( FALSE );
   if ( NOT (point=(POINT_T *)node->data)) return(FALSE);

   SetAPen(window->RPort,3);          /* highlight it  */
   (void) displayPoint( window,node );

   /* put data into strings & display message window   */

   sprintf(str1,”Node = %x, next = %x, data = %x”,
                node,node->next,node->data);
   sprintf(str2,”Picked entity type is POINT_TYPE”);
   sprintf(str3,”x = %d, y = %d”,point->x,point->y);

   gpDisplayMsg(str1,str2,str3);    /* display the data */

   SetAPen(window->RPort,1);        /* unhighlight it  */
   (void) displayPoint( window,node );
   return( TRUE );
}
/*======================================================*/
/*
 *  Procedures for Lines:
 *  createLine() - create and initialize a line & its
 *                 list node
 *  deleteLine() - delete an instance of a line
 *  reportLine() - output info about a line instance
 *  searchLine() - see if a line is at input coordinates
 *  displayLine() - graphically display a line segment
 *  displayLineInfo() - textually display line info
*/
/*======================================================*/

NODE_T *createLine( x1,y1,x2,y2 )
short    x1,y1;
short    x2,y2;
{
   NODE_T   *node;   /* the list node to carry the line */
   LINE_T   *line;   /* the line structure              */

   if ( NOT (node=NEW(NODE_T)) )  /* allocate the node   */
      return( NULL );

   if ( NOT (line=NEW(LINE_T)) )  /* allocate the line   */
   {                              /* didn't get it       */
      FREE( node );               /* release node memory */
      return( NULL );
   }
   line->x1 = x1;  line->y1 = y1; /* init the line       */
   line->x2 = x2;  line->y2 = y2;

   /* initialize the node.  Coerce line to unsigned char */

   node->next = NULL;
   node->dataType = LINE_TYPE;
   node->data = (unsigned char*)line;
   return( node );
}
/*---------------------------*/
BOOLEAN deleteLine( node )
NODE_T  *node;
{
   LINE_T  *line;

   /*
    *  make sure we have a valid node pointer & correct
    *  data type
   */
   if ( NOT node || node->dataType != LINE_TYPE )
      return( FALSE );
   if ( NOT (line = (LINE_T*)node->data) )
      return( FALSE );

   FREE( line );
   FREE( node );
   return( TRUE );
}
/*---------------------------*/
BOOLEAN reportLine( outFile,node )
FILE     *outFile;
NODE_T   *node;
{
   LINE_T  *line;

   /*
    *  make sure pointer is valid  & type is correct
   */
   if ( NOT node || node->dataType != LINE_TYPE )
      return( FALSE );
   if ( NOT (line = (LINE_T*)node->data) )
      return( FALSE );

   fprintf(outFile,”>  line = %x, x1 = %d, y1 = %d”,
           line,line->x1,line->y1);
   fprintf(outFile,” x2 = %d, y2 = %d\n”,
           line->x2,line->y2);

   return( TRUE );
}
/*---------------------------*/
BOOLEAN searchLine( node,x,y )
NODE_T  *node;
short   x,y;
{
   LINE_T  *line;
   short   minX,minY,maxX,maxY;
   short   eps = 4;

   if ( NOT node || NOT (line = (LINE_T*)node->data) )
      return( FALSE );

   /*
    *  NOTE:  this method of determining if point is on
    *  line only works for horiz & vert line segments
   */

   /*  build an ‘epsilon’ box around line segment        */

   minX = MIN(line->x1,line->x2) - eps;
   minY = MIN(line->y1,line->y2) - eps;
   maxX = MAX(line->x1,line->x2) + eps;
   maxY = MAX(line->y1,line->y2) + eps;

   if ( minX <= x && x <= maxX &&
        minY <= y && y <= maxY    )
      return( TRUE );

   return( FALSE );
}
/*---------------------------*/
BOOLEAN displayLine( window,node )
struct Window *window;
NODE_T        *node;
{
   LINE_T *line;

   /* make sure we have valid data, then display it    */

   if ( NOT window || NOT node ) return( FALSE );
   if ( NOT (line = (LINE_T *)node->data)) return(FALSE);

   Move(window->RPort,line->x1,line->y1);
   Draw(window->RPort,line->x2,line->y2);
   return( TRUE );
}
/*---------------------------*/
BOOLEAN displayLineInfo( window,node )
struct Window *window;
NODE_T        *node;
{
   LINE_T *line;
   char    str1[61],str2[61],str3[61];

   /* make sure we have valid data, then display it    */

   if ( NOT window || NOT node ) return( FALSE );
   if ( NOT (line = (LINE_T *)node->data)) return(FALSE);

   SetAPen(window->RPort,3);          /* highlight line */
   (void) displayLine( window,node );

   /* put data into strings & display message window   */

   sprintf(str1,”Node = %x, next = %x, data = %x”,
                node,node->next,node->data);
   sprintf(str2,”Picked entity type is LINE_TYPE”);
   sprintf(str3,”x1 = %d, y1 = %d, x2 = %d, y2 = %d”,
           line->x1,line->y1,line->x2,line->y2);

   gpDisplayMsg(str1,str2,str3);    /* display the data */

   SetAPen(window->RPort,1);        /* unhighlight line */
   (void) displayLine( window,node );
   return( TRUE );
}
/*======================================================*/
/*
 *  Intuition Utility Procedures:
 *  openLibs() - open libraries
 *  closeLibs() - close intuition libraries
 *  allocWindow() - allocate,inititalize & open window
 *  gpDisplayMsg() - display a message to a graphical
 *                   message window
 *  handleInput - intuition input handler
```

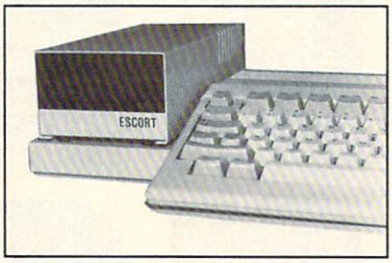
```
*/
/*=====================================================*/

BOOLEAN openLibs()
{
   IntuitionBase = (struct IntuitionBase *)
            OpenLibrary("intuition.library",INTUITION_REV);

   if ( NOT IntuitionBase ) return( FALSE );

   GfxBase = (struct GfxBase *)
              OpenLibrary("graphics.library",GRAPHICS_REV);

   if ( NOT GfxBase ) return( FALSE );
   return( TRUE );
}
/*-----------------------------*/
void closeLibs()
{
   if ( IntuitionBase ) CloseLibrary(IntuitionBase);
   if ( GfxBase ) CloseLibrary(GfxBase);
}
/*-----------------------------*/
struct Window *allocWindow( screen,gadget,x,y,w,h,name,
                            wFlags,iFlags,c0,c1 )
struct Screen  *screen;
struct Gadget  *gadget;
SHORT          x,y,w,h;
char           *name;
ULONG          wFlags;
USHORT         iFlags;
SHORT          c0,c1;
{
   struct NewWindow  NewWindow;
   struct Window     *window;

   NewWindow.LeftEdge = x;
   NewWindow.TopEdge = y;
   NewWindow.Width = w;
   NewWindow.Height = h;
   NewWindow.DetailPen = c0;
   NewWindow.BlockPen = c1;
   NewWindow.Title = name;
   NewWindow.Flags = wFlags;
   NewWindow.IDCMPFlags = iFlags;
   NewWindow.Type = WBENCHSCREEN;
   NewWindow.FirstGadget = gadget;
   NewWindow.CheckMark = NULL;
   NewWindow.Screen = screen;
   NewWindow.BitMap = NULL;
   NewWindow.MinWidth = 20;
   NewWindow.MinHeight =20;
   NewWindow.MaxWidth = 640;
   NewWindow.MaxHeight = 400;
   window = (struct Window *)OpenWindow( &NewWindow );
   return( window );
}
/*-----------------------------*/
void gpDisplayMsg(str1,str2,str3)
char *str1,*str2,*str3;
{
   struct IntuiMessage *msg;
   struct Window       *msgWindow;
   unsigned long       wFlags,class;
   unsigned short      iFlags=CLOSEWINDOW|INACTIVEWINDOW;

   if ( *str1 == '\0' && *str2 == '\0' && *str3 == '\0' )
      return;

   /*  allocate the message window                      */

   wFlags = WINDOWCLOSE|WINDOWDRAG|ACTIVATE;
   if ( NOT (msgWindow = allocWindow(NULL,NULL,50,20,550,
                  70,"= MESSAGE =",wFlags,iFlags,-1,-1)) )
      return;

   if ( strlen(str1) > 60 ) /* clip txt len to 60 chars  */
      *(str1+60) = '\0';
   if ( strlen(str2) > 60 )
      *(str2+60) = '\0';
   if ( strlen(str3) > 60 )
      *(str3+60) = '\0';

   SetAPen(msgWindow->RPort,1);  /*  display messages    */
   Move(msgWindow->RPort,10,20);
   Text(msgWindow->RPort,str1,strlen(str1));
   Move(msgWindow->RPort,10,30);
   Text(msgWindow->RPort,str2,strlen(str2));
   Move(msgWindow->RPort,10,40);
   Text(msgWindow->RPort,str3,strlen(str3));

   /* wait for message, then clear port & close window   */

   for(;;)
   {
      Wait( 1 << msgWindow->UserPort->mp_SigBit );

      while( msg = (struct IntuiMessage*)
                   GetMsg(msgWindow->UserPort) )
      {
         class = msg->Class;
         ReplyMsg( msg );

         if ( class == CLOSEWINDOW ||
              class == INACTIVEWINDOW )
         {
            while( msg = (struct IntuiMessage*)
                   GetMsg(msgWindow->UserPort) )
               ReplyMsg( msg );

            CloseWindow( msgWindow );
            return;
         }  /*  end if  */
      }  /*  end while  */
   }  /*  end for       */
}
/*-----------------------------*/
void handleInput( window,hdr )
struct Window *window;
HEADER_T      *hdr;
{
   struct IntuiMessage *message;
   NODE_T              *node;
   unsigned short      mCode;
   unsigned long       mClass,iFlags;
   short               mX,mY;

   iFlags = window->IDCMPFlags; /* save IDCMP flags      */


   /* wait for a message to arrive                       */

   for (;;)
   {
      Wait( 1 << window->UserPort->mp_SigBit );

      /* process messages until there are none           */

      while( (message = (struct IntuiMessage *)
                        GetMsg(window->UserPort) ) )
      {
         /* get message info we'll need and process it   */

         mClass = message->Class;  mCode = message->Code;
         mX = message->MouseX;  mY = message->MouseY;
         ReplyMsg( message );

         /* see if this is an event we're interested in  */

         if ( mClass == CLOSEWINDOW )  /* shutdown & stop*/
         {
            while( (message = (struct IntuiMessage *)
                              GetMsg(window->UserPort) ) )
               ReplyMsg( message );

            ModifyIDCMP(window,NULL);     /* halt input  */
            return;
         }
         if ( mClass == MOUSEBUTTONS && mCode == SELECTUP )
         {
            /* search for entity & show entity info      */

            if ( node = searchList(hdr,mX,mY) )
            {
               ModifyIDCMP(window,NULL);  /* block input */
               (void) displayNodeInfo(window,node);
               ModifyIDCMP(window,iFlags);/* resume input*/
            }
         }
      }  /*  end while  */
   }  /* end for     */
}
```

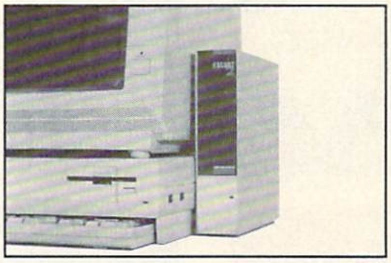
•AC•

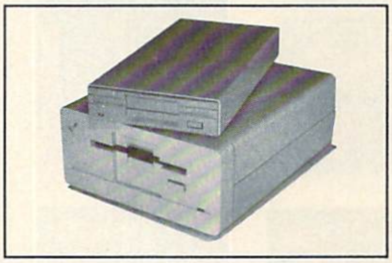
POWER HUNGRY



Whether you own an Amiga 500, 1000 or 2000 you'll want to feed it right. Fortunately we're here to help. At Expansion Technologies we provide high quality expansion products that are guaranteed to keep your Amiga growing happy and strong. Just check out our menu :







If your Amiga has a really voracious appetite, but your pocket book is on a diet, try our Escort 30Mb SCSI hard drive at a 20Mb price. Controllers, with a bus return are available for both the A500 as well as the A1000. Each accepts our 3.5 or 5.25 inch drive chassis (available in sizes up to 80Mb) with fan and power supply and is fully compatible with WorkBench 1.3. If you prefer buffet, buy the kit and supply your own drive. However you do it you'll enjoy lots of storage space at a great price.
Now thats serious fast food!

Escort Hard Drive kits

A500 30Mb	**$699**
A500 Kit 3.5 inches	**$339**
A1000 30Mb	**$759**
A1000 Kit 3.5 inches	**$379**

Call for other sizes.

For the discriminating pallet of the expansion hungry Amiga 1000 we have the ever popular Escort 2. It's 2 megabytes of no wait-state memory that meets all the known standards and then some. It's also a uniquely designed vertical 2-slot card cage that offers incredible flexibility. Like the ability to upgrade to a scrumptious 4 megabytes of memory, or you can add a hard disk controller that uses IBM type hard drives, or an external power supply for the bus return or...well, you get the idea.
Truly a meal fit for a king!

Escort 2 (0k mem)	**$339**
2 megabyte upgrade card (0K mem)	**$279**
ST-506 hard drive controller card and power supply	**$399**

Simply the best engineered 3.5 inch external floppy drive available for your Amiga, and it's 100% 1010 compatible. The precision drive mechanism is ultra quiet and has the highest MTBF specifications. A low power CMOS design and pass through allows for easy daisy chaining too. Its low profile (1x4x8 in.) allows A500 disks to eject over the top, and it fits right under an A1000. Includes an extra long cable, heavy steel chassis and faceplate with dust cover. Why buy less than this Made In The U.S.A drive.
Now you're cooking with gas!

Escort 3.5 in. floppy	**$179**

**Feed your power hungry Amiga right-
give it peripherals from Expansion Technologies.**

• Toothpicks not included

VISA and MasterCard accepted. Dealer Inquiries Invited.

Lunch for Lawyers: Amiga is a trademark of Commodore-Amiga, Inc. IBM is a trademark of International Business Machines Corporation. Seagate is a trademark of Seagate, Inc. Escort is a trademark of Expansion Technologies Inc.

EXPANSION
TECHNOLOGIES

46127 Landing Pkwy, Fremont, CA 94538 415-656-2890

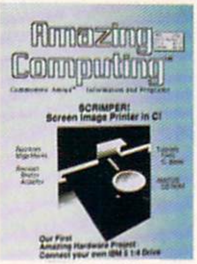
Amazing COMPUTING™

• EXPANDING REFERENCE •

VOLUME 1.1

VOLUME 1.2

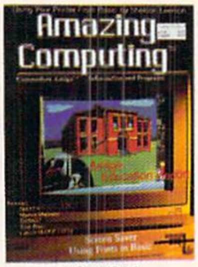
VOLUME 1.3


VOLUME 1.4

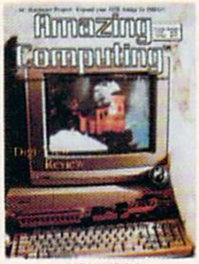
VOLUME 1.5

VOLUME 1.6

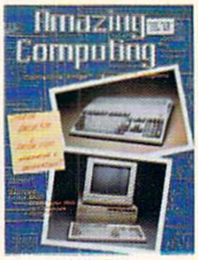
VOLUME 1.7

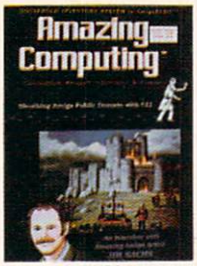

VOLUME 1.8

VOLUME 1.9


VOLUME 2.1

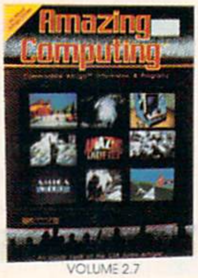
VOLUME 2.2


VOLUME 2.3

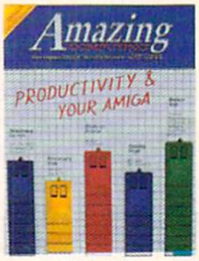

VOLUME 2.4

VOLUME 2.5

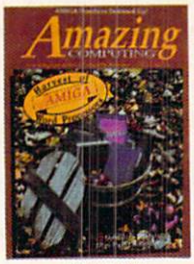
VOLUME 2.6

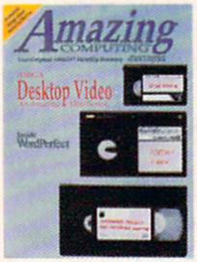

VOLUME 2.7

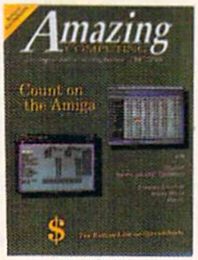
VOLUME 2.8

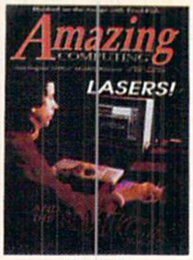

VOLUME 2.9

VOLUME 2.10


VOLUME 2.11

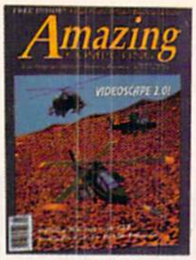

VOLUME 2.12


VOLUME 3.1


VOLUME 3.2

VOLUME 3.3

VOLUME 3.4


VOLUME 3.5

VOLUME 3.6

VOLUME 3.7

VOLUME 3.8

VOLUME 3.9

Your Resource to the Commodore Amiga

Expanding reference is not just an empty promise. The pages of Amazing Computing™ are filled with articles on technical operations and procedures, basic use, and just-plain-fun. The growing library of Amazing Computing's Back Issues contains articles ranging from building your own IBM Disk controller, to setting up your own startup sequence. Amazing Computing™ has repeatedly been the first magazine to offer the Amiga users solid, in depth reviews and hands on articles for their machines.

From the Beginning

Since February 1986, Amazing Computing™ has been providing users with complete information for their Amigas. This store house of programs and information is still available through our back issues. From the Premiere issue to the present, there are insights into the Amiga any user will find useful. AC was the first magazine to document CLI, tell its readers how to connect a 5 1/4 IBM drive, describe a 1 meg upgrade hardware project for the A1000, and many more. Please read the list of topics AC has covered below to find the information you have been missing.

Back Issues are $5.00 US, $6.00 Canada and Mexico, $7.00 Foreign Surface

All payments must be made by check or money order in U.S. funds drawn on a U.S. Bank.

Limited Supply

Unfortunately, nothing lasts forever, and the availability of some of our Back Issues is definitely limited. Complete your Amazing Computing™ library today, while these issues are still available, by completing the order form in the back of this issue.

Volume 1 Number 1 Premiere 1986

Super Spheres By Kelly Kauffman An ABasic Graphics prog.
Date Virus By J Foust A disease may attack your Amiga!
EZ-Term by Kelly Kauffman An ABasic Terminal program
Miga Mania by P. Kivolowitz Programming fixes & mouse care
Inside CLI by G. Musser a guided insight into the AmigaDos™
CLI Summary by G. Musser Jr. A list of CLI commands
AmigaForum by B. Lubkin Visit Compuserve's Amiga SIG
Commodore Amiga Development Program by D. Hicks
Amiga Products A listing of present and expected products

Volume 1 Number 2 March 1986

Electronic Arts Comes Through A review of software from EA
Inside CLI: part two G. Musser Investigates CLI & ED
A Summary of ED Commands
Live! by Rich Miner A review of the Beta version of Live!
Online and the CTS Fabite 2424 ADH Modem by J. Foust
Superterm V 1.0 By K. Kauffman A term. prog. in Amiga Basic
A Workbench "More" Program by Rick Wirch
Amiga BBS numbers

Volume 1 Number 3 April 1986

Analyze! a review by Ernest Viverios
Reviews of Racter, Barataccas and Mindshadow
Forth! The first of our on-going tutorial
Deluxe Draw!! by R. Wirch An Amiga Basic art program
Amiga Basic, A beginners tutorial
Inside CLI: part 3 by George Musser George gives us PIPE

Volume 1 Number 4 May 1986

SkyFox and Articfox Reviewed
Build your own 5 1/4 Drive Connector By Ernest Viveiros
Amiga Basic Tips by Rich Wirch
Scrimper Part One by P. Kivolowitz prog to print Amiga screen
Microsoft CD ROM Conference by Jim O'Keane
Amiga BBS Numbers

Volume 1 Number 5 1986

The HSI to RGB Conversion Tool
by S. Pietrowicz Color manipulation in BASIC
AmigaNotes by Rick Rae The first of the Amiga music columns
Sidecar A First Look by John Foust A first "under the hood"
John Foust Talks with R. J. Mical at COMDEX™
How does Sidecar affect the Transformer
an interview with Douglas Wyman of Simile
The Commodore Layoffs by J. Foust A look Commodore "cuts"
Scrimper Part Two by Perry Kivolowitz
Marauder reviewed by Rick Wirch
Building Tools by Daniel Kary

Volume 1 Number 6 1986

Temple of Apshai Triology reviewd by Stephen Pietrowicz
The Halley Project: A Mission in our Solar System
reviewed by Stephen Pietrowicz
Flow: reviewed by Erv Bobo
Textcraft Plus a First Look by Joe Lowery
How to start your own Amiga User Group by William Simpson
Amiga User Groups
Mailing List by Kelly Kauffman a basic mail list program
Pointer Image Editor by Stephen Pietrowicz
Scrimper: part three by Perry Kivolowitz
Fun With the Amiga Disk Controller by Thom Sterling
Optimize Your AmigaBasic Programs for Speed by Pietrowicz

Volume 1 Number 7 1986

Aegis Draw: CAD comes to the Amiga by Kelly Adams
Try 3D by Jim Meadows an introduction to 3D graphics
Aegis Images/ Animator: a review by Erv Bobo
Deluxe Video Construction Set reviewed by Joe Lowery
Window requesters in Amiga Basic by Steve Michel
ROT by Colin French a 3D graphics editor
"I C What I Think" Ron Peterson with a few C graphic progs
Your Menu Sir! by B Catley program Amiga Basic menues
IFF Brush to AmigaBasic 'BOB' Basic editor by M Swinger
Linking C Programs with Assembler Routines
on the Amiga by Gerald Hull

Volume 1 Number 8 1986

The University Amiga By G.Gamble
Amiga's inroads at Washington State University
MicroEd a look at a one man army for the Amiga
MicroEd, The Lewis and Clark Expedition reviewed Frizelle
Scribble Version 2.0 a review
Computers in the Classroom by Robert Frizelle
Two for Study by Frizelle Discovery & TheTalking Coloring Book
True Basic reviewed by Brad Grier
Using your printer with the Amiga
Marble Madness reviewed by Stephen Pietrowicz
Using Fonts from AmigaBasic by Tim Jones
Screen SaVer by P. Kivolowitz A monitor protection prog. in C
Lattice MAKE Utility reviewed by Scott P. Evernden
A Tale of Three EMACS by Steve Poling
.bmap File Reader in Amiga Basic by T Jones

Volume 1 Number 9 1986

Instant Music Reviewed by Steve Pietrowicz
Mindwalker Reviewed by Richard Knepper
The Alegra Memory Board Reviewed by Rich Wirch
TxEd Reviewed by Jan and Cliff Kent
Amazing Directory A guide to the sources and resources
Amiga Developers A listing of Suppliers and Developers
Public Domain Catalog A listing of Amicus and Fred Fish PDS
Dos 2 Dos review R. Knepper
Transfer files from PC/MS-DOS and AmigaBasic
MaxiPlan review by Richard Knepper The Amiga Spreadsheet
Gizmoz by reviewed by Peter Wayner Amiga extras!
The Loan Information Program by Brian Catley
basic prog. to for your financial options
Starting Your Own Amiga Related Business by W. Simpson
Keep Track of Your Business Usage for Taxes by J. Kummer
The Absoft Amiga Fortran Compiler reviewed by R A. Reale
Using Fonts from AmigaBasic, Part Two by Tim Jones
68000 Macros on the Amiga by G. Hull Advance your ability.
TDI Modla-2 Amiga Compiler review by S Faiwisze

Volume 2 Number 1 1987

What Digi-View Is... Or, What Genlock Should Be! by J. Foust
AmigaBasic Default Colors by Bryan Catley
AmigaBasic Titles by Bryan Catley
A Public Domain Modula-2 System reviewed by Warren Block
One Drive Compile by Douglas Lovell
Using Lattice C with a single drive system
A Megabyte Without Megabucks by Chris Irving
An Internal Megabyte upgrade
Digi-View reviewed by Ed Jakober
Defender of the Crown reviewed by Keith Conforti
Leader Board reviewed by Chuck Raudonis
Roundhill Computer System's PANEL reviewed by Ray Lance
Digi-Paint.....by New Tek previewed by John Foust
Deluxe Paint II ....from Electronic Arts previewed by J. Foust

Volume 2 Number 2 1987

The Modem by Josph L. Rothman efforts of a BBS Sysop
MacroModem reviewed by Stephen R. Pietrowicz
GEMINI or "It takes two to Tango" by Jim Meadows
Gaming between machines
BBS-PC! reviewed by Stephen R. Pietrowicz
The Trouble with Xmodem by Joseph L. Rothman
The ACO Project....Graphic Teleconferencing on the Amiga
by S. R. Pietrowicz
Flight Simulator II....A Cros Country Tutorial by John Rafferty
A Disk Librarian in AmigaBASIC by John Kennan
Creating and Using Amiga Workbench Icons by C. Hansel
AmigaDOS version 1.2 by Clifford Kent
The Amazing MIDI Interface build your own by Richard Rae
AmigaDOS Operating System Calls and
Disk File Management by D. Haynie
Working with the Workbench by Louis A. Mamakos Prog in C

Volume 2 Number 3

The Amiga 2000™ by J Foust
A First look at the new, high end Amiga™
The Amiga 500™ by John Foust
A look at the new, low priced Amiga
An Analysis of the New Amiga PCs by J. Foust
Speculation on the New Amigas
Gemini Part II by Jim Meadows
The concluding article on two-player games
Subscripts and Superscripts in AmigaBASIC by Ivan C. Smith
The Winter Consumer Electronics Show by John Foust
AmigaTrix by W. Block Amiga™ shortcuts
Intuition Gadgets by Harriet Maybeck Tolly
A journey through gadget-land, using C
Shanghai reviewed by Keith M. Conforti
Chessmaster 2000 & Chessmate reviewedby Edwin V. Apel, Jr.
Zing! from Meridian Software reviewed by Ed Bercovitz
Forth! by Jon Bryan Get stereo sound into your Forth programs.
Assembly Language on the Amiga™ by Chris Martin
Roomers by theBandito Genlocks are finally shipping, & MORE!!!
AmigaNotes by R. Rae Hum Busters... "No stereo? Y not?...
The AMICUS Network by J. Foust
CES, user group issues and Amiga Expo"

Volume 2 Number 4 1987

Amazing Interviews Jim Sachs by S. Hull Amiga Artist
The Mouse That Got Restored by Jerry Hull and Bob Rhode
Sluething Public Domain Disks with CLI by John Foust
Highlights: the San Francisco Commodore Show by S Hull
Speaker Sessions: San Francisco Commodore Show H Tolly
Household Inventory System in AmigaBASIC™ by B Catley
Secrets of Screen Dumps by Natkun Okun
Using Function Keys with MicroEmacs by Greg Douglas
Amigatrix II by Warren Block More Amiga shortcuts
Basic Gadgets by Brian Catley Create gadget functions
Gridiron reviewed by K. Conforti Real football for the Amiga
Star Fleet I Version 2.1 reviewed by J. Tracy Amigain Space
The TIC reviewed by J. Foust Battery powered Clock Calendar
Metascope review by H. Tolly An easy-to-use debugger

Volume 2 Number 5 1987

The Perfect Sound Digitizer review by R. Battle
The Future Sound Digitizer by W. Block Applied Vision's SD
Forth! by J. Bryancomparing JForth amd Multi-Forth.
Basic Input by B. Catley AmigaBASIC input routine for use in
all your programs.
Writing a SoundScape Module in C by T. Fay Programming with MIDI, Amiga and SoundScape by SoundScape author.

Volume 2 Number 5 1987 continued

Programming in 68000 Assembly Language by C. Martin
Continuing with Counters & Addressing Modes.
Using FutureSound with AmigaBASIC by J. Meadows
AmigaBASIC Programming utility with real, digitized STEREO
AmigaNotes Rich Rae reviews SoundScape Sound Sampler.
More AmigaNotes by R. Rae A furtherlook at Perfect Sound .
Waveform Workshop in AmigaBASIC by J. Shields edit & save
waveform for use in other AmigaBASIC programs.
The Mimetics Pro MIDI Studio by Sullivan, Jeffery
A review of Mimetics' music editor/player.
Intuition Gadgets Part II by H. MaybeckTolly Boolean gadgets
provide the user with an on/off user interface.

Volume 2 Number 6 1987

Forth! by J. Bryan Access resources in the ROM Kernal.
The Amazing Computing Hard Disk Review by J. Foust & S. Leemon In-depth looks at the C Ltd. Hard Drive, Microbotics' MAS-Drive20, Byte by Byte's PAL Jr., Supra's 4x4 Hard Drive and Xebec's 9720H Hard Drive. Also. a look at disk driver software currently under devlopment.

Modula-2 AmigaDOS™ Utilities by S. Faiwiszewsk A
Calls to AmigaDOS and the ROM kernal.
Amiga Expansion Peripheral by J. Foust
Explanation of Amiga expansion peripherals.
Amiga Technical Support by J. Foust
How and where to get Amiga tech support.
Goodbye Los Gatos by J. Foust Closing Los Gatos.
The Amicus Network by J. Foust West Coast Computer Faire.
Metacomco Shell and Toolkit by J. Foust A review
The Magic Sac by J. Foust Run Mac programs on your Amiga.
What You Should Know Before Choosing an Amiga 1000
Expansion Device by S. Grant
7 Assemblers for the Amiga by G. Hull Choose your assembler
Shakeup Replaces Top Management at Commodore by S. Hull
Peter J. Baczor"by S. Hull Manager at CBM gives an inside look
Logistix A review by Richard Knepper
Organize! by A review Richard Knepper database.
68000 Assembly Language Programming on the Amiga
by Chris Martin
Superbase Personal Relational Database by Ray McCabe
AmigaNotes by Rae, Richard A look at FutureSound
Commodore Shows the Amiga 2000 and 500 at the Boston
Computer Society by H Maybeck Tolly

Volume 2, Number 7 1987

New Breed of Video Products by John Foust...
Very Vivid! by Tim Grantham...
Video and Your Amiga by Oran Sands III
Amigas & Weather Forecasting by Brenden Larson
A-Squared and the Live! Video Digitizer by John Foust.
Aegis Animator Scripts and Cel Animation by John Foust
Quality Video from a Quality Computer by Oran Sands III.
Is IFF Really a Standard? by John Foust..
Amazing Stories and the Amiga™ by John Foust.
All about Printer Drivers by Richard Bielak
Intuition Gadgets by Harriet Maybeck Tolley.
Deluxe Video 1.2 by Bob Eller
Pro Video CG1 by Oran Sands III.
Digi-View 2.0 Digitizer/Software by Jennifer M. Janik
Prism HAM Editor from Impulse by Jennifer M. Janik
Easyl drawing tablet by John Foust..
CSA's Turbo-Amiga Tower by Alfred Aburto
68000 Assembly Language by Chris Martin.

Volume 2, Number 8 1987

This month Amazing Computing™ focuses on entertainment packages for the Amiga. Amazing game reviews...

SDI, Earl Weaver Baseball, Portal, The Surgeon, Little Computer People, Sinbad, StarGlider, King's Quest I,II and III, Faery Tale Adventure, Ultima III, Facets of Adventure, Video Vegas and Bard's Tale.

Plus Amazing monthly columns... Amiga Notes, Roomers, Modula-2, 68000 Assembly Language and The Amicus Network.
Disk-2-Disk by Matthew Leeds
The ColorFonts Standard by John Foust
Skinny C Programs by Robert Riemersma, Jr.
Hidden Messages In Your Amiga™ by John Foust
The Consumer Electronics Show and Comdex.by J Foust

Volume 2 Number 9 1987

Analyze 2.0 reviewed by Kim Schaffer
Impact Business Graphics review by Chuck Raudonis
Microfiche Filer review by Harv Laser
Pagesetter review by Rick Wirch
Gizmoz Productivity Set 2.0 review by Bob Eller
Kickwork review by Harv Laser
Diga Telecommunications Package review by Steve Hull
Mouse Time and Timesaver review by John Foust
Insider Memory Expansion review by James O'Keane
Microbotics Starboard-2 review by S. Faiwiszewski
Leather Goddesss of Phobos
reviewed by Harriet Maybeck-Tolly
Lattice C Compiler Version 3.10 reviewed by Gary Sarff
Manx 3.4a Update reviewed by John Foust
AC-BASIC reviewed by Sheldon Leemon
AC-BASIC Compilier an alternative comparison by B Catley
Modula-2 Programming S Faiwiszewski
Raw Console Device Events
Directory Listings Under AmigaDOS by Dave Haynie
AmigaBASIC Patterns by Brian Catley
Programming with Soundscape
Todor Fay manipulate's samples
Bill Volk, Vice-President Aegis Development,
interviewed by Steve Hull
Jim Goodnow, Devekoper of Manx 'C' interview by Harriet M Tolly
Plus a great collection of monthly columns...

Volume 2 Number 10 1987

Max Headroom and the Amiga by John Foust
Taking the Perfect Screen Shot by Keith Conforti
Amiga Artist: Brian Williams by John Foust
Amiga Forum on Compuserve™... Software Publishing Conference Transcript by Richard Rae
All About Online Conferencing by Richard Rae
dBMAN reviewed by Clifford Kent
Amiga Pascal reviewed by Michael McNeil
AC-BASIC Compiler reviewed by Bryan Catley
68000 Assembly Language by Chris Martin
Amiga Programming:
Amiga BASIC Structures by Steve Michel
Quick and Dirty Bobs by Michael Swinger
Directory Listings Under Amiga-DOS, Part II by Dave Haynie
Fast File I/O with Modula-2 by Steve Faiwiszewski
Window I/O by Read Predmore
Plus a great collection of monthly columns...

Volume 2 Number 11 1987

Word Processors Rundown by Geoff Gamble
ProWrite, Scribble!, and WordPerfect compared
LPD Writer Review by Marion Deland
VizaWrite Review by Harv Laser
Aedit Review by Warren Block
WordPerfect Preview by Harv Laser
Jez San Interview by Ed Bercovitz—StarGlider author speaks!
Do-it-yourself Improvements to the Amiga Genlock
Digi-Paint Review by Harv Laser
Sculpt 3D Review by Steve Pietrowicz
Shadowgate Review by Linda Kaplan
TeleGames Review by Michael T. Cabral
Reason Preview: an intense grammar examination application
As I See It by Eddie Churchill Peeking at WordPerfect,
Gizmoz V2.0 and Zing! Keys
AmigaNotes by R Rae 4 electronic music books
Modula-2 Programming by Steve Faiwiszewski
devices, I/O, and the serial port
68000 Assembly Language by Chris Martin
Chris walks through the display routines
The AMICUS Network by John Foust—Desktop Publishing, Seybold
C Animation Part II by Mike Swinger Animation Objects
BASIC Text by Brian Catley Pixel perfect text positioning
Soundscape Part III by Todor Fay VU Meter and more
Fun with Amiga Numbers by Alan Barnett
File Browser by Bryan Catley—Full Feature BASIC File Browsing
Plus a great collection of monthly columns...

Volume 2 Number 12 1987

The Ultimate Video Accessory by Larry White
The Sony Connection by Stewart Cobb
15-Puzzle in AmigaBASIC by Zoltan Szepsi
Life, Part I: The Beginning by Gerald Hull
The ultra-complex nine blit solution to the "Game of Life."
Amiga Virus! by John Foust
A new Amiga virus has surfaced. Please check your system.
CLI Arguments in C by Paul Castonguay
MIDI Interface Adapter by Barry Massoni
Amiga 1000-style MIDI interfaces can fit A2000s or 500s
Modula-2 by Steve Faiwiszewski
First in a series, a command line calculator in Modula-2.
AmigaNotes by Richard Rae
The audio changes made in the Amiga 500 and 2000.
Animation for C Rookies: Part III by M. Swinger
tackling double-buffering.
The Big Picture by Warren Ring
Amiga™ Assembly language programming for the brave!
Karate Kid Reviewby Stephen R. Pietriowicz
GO! 64 review by John Foust, James O'Keane, and Rick Wirch
Three C-64 experts investigate a new Amiga 64 emulator.
A-Talk-Plus Review by Brendan Larson
"Full-fledged terminal program" & Tektronics capabilities
Calligrapher Review by John Foust
Animator: Apprentice Review by John Foust
Playing Dynamic Drums on the Amiga by David N. Blank
WordPerfect Review by Steve Hul
Insider/Kwikstart Review by Ernest P. Viveiros Sr
RAM & ROM expansion: Comments and installation tips.
Bug Bytes by John Steiner
Forth! by Jon Bryan
DumpRPort utility for your Multi-Forth toolbox.
As I See It by Eddie Churchill
An offbeat look on Digi-Paint, Portal, and Videoscape 3D.
The Amicus Network by John Foust
The Commodore Show and AmiExpo: New York!

Volume 3 Number 1 1988

AmigaNotes by **Richard Rae**
Digital music generation on tha Amiga .
C Animation Part IV by Michael Swinger
Just when you thought it was safe to go back in the C waters...
Forth by John Bryan
Sorting out CHIP and FAST memory on the Amiga.
The Big Picture by Warren Ring Daring assembler language programming; CLI system calls and manipulating disk files.
Bug Bytes by John Steiner
Roomers by The Bandito
Amiga Dos1.3? 80386-based BridgeBoard for the A2000? More!
As I See It by Eddie Churchill
Opinions, obsevations, & the birth of a new software generation.
68000 ASssembly Langueage Programming by Chris Martin
"Create a multi-color screen without using Intuition routines!"
Modula-2 Programming by Steve Faiwiszewski
A new contender bursts on to the modula-2 scene!
Amicus Network Special Report: Fall COMDEX by J. Foust
Commodore at COMDEX and new products
The ultimate Video Accessory: Part II by Larry White
Life: Part II by Gerald Hull
"A detailed look at efficient use of the Amiga blitter."
FormatMaster: Professional Disk Formatting Engine byC.Mann
Put Batch language to work on the drudgery of disk formatting .
BSpread by Brian Catley
A full featured AmigaBASIC spreadsheet you can program!
AmigaForum Transcript ed. by Richard Rae
Zoom in on Commodore/ Amiga's Dave Haynie.
Haicalc Review by Chuck Raudonis
"A straightforward, easy to use, functional spreadsheet."
VIP Professional Review by Suzanne Mitchell
Easy stock porfolio management on the Amiga.
Money Mentor Review by Stephen Kemp
A personal finance system beyond your checkbook.
Investor's Advantage Review by Richard Knepper
plus "Poor Man's Guide to the Stock Market."

Volume 3 Number 2 1988

Laser Light Shows with the Amiga by Patrick Murphy
Lasers and the Amiga: A Dazzling Tandem
The Ultimate Video Accessory: PartIII by Larry White
Take the final stepstoward designing your own videos.
Our First Desktop Video by Larry White
Step-by -step guide to organizing & presenting your Amiga video.
Hooked on the Amiga with Fred Fish by Ed Berkovitz
Inside views from the man behind all those "Fish" disks.
Photo Quality Reproduction with the Amiga and Digi-View
by Stephen Lebans
Balancing your Checkbook with WordPerfect Macros by S.Hull
Hand your checkbook worries over to the Amiga.
More Basic Text by Bryan Catley easier text on an Amiga screen
Life: Part III by Gerald Hull
Sries winds up with famed nine-blit calculation & source to LIFER.
Solutions to Linear Algebra through Matrix Computations
by Ropbert Ellis
Simplify matrix algebra with basic operations & routines.
Roomers by Bandito Amiga 3000, Virus news,& Laser Toaster
Bug Bytes by John Steiner
Modula-2 Programming by Steve Faiwiszewski
Catching up with Calc-a source follow-up.
68000 Assembler Language Programming by Chris Martin
Graphics- Part II of Assemgram.
Arazok's Tomb by Kenneth E. Schaefer
"A terrifyig adventure into the world of the occult."
AIRT by Steve Faiwiszewski
An innovative icon-basede programming language.
Forms in Flight by Steve Pietrowicz
Render and Animate objects in 3D!
Silicon Dreams and the Jewel of Darkness by K E. Schaefer
Leisure suit Larry by Kenneth E. Schaeler
Two New Entries From Microbiotics by John Foust
M501 Expansion & Starboard II MultiFunctionboard.
Mindlight 7 and People Meter by John Foust
Phantasie
Ken E. Schaefer with the Amazing Phantasie Character Editor.

Volume 3 Number 3 1988

Take Five! by Steve Hull
Blast your frustrations in these five Amiga games.
Desktop Video,Part IV by Larry White
Put all the pieces together-the desktop video commercial.
The Hidden Power of CLI Batch File Processing by J. Rothman
Make your Amiga easier to use with CLI Batch files.
A Conference with Eric Graham edited by John Foust
The mastermind behind Sculpt 3D and Animate 3D.
Perry Kivolowitz Interviewed by Ed Bercovitz
Amiga insights from a major developer and personality.
Jean "Moebius" Giraud Interviewed by Edward L. Fadigan
Avant-garde art comes to the Amiga-in dazzling form.
PAL Help by Perry Kivolowitz
All the help you need for a A1000 expansion reliability.
Boolean Function Minimization by Steven M. Hart
A useful digital design tool in AmigaBASIC.
Amiga Serial Port and Midi Compatibility for Your A2000!
by Lynn Ritter and Gary Rentz
Add an A1000-style serial port to your A2000!
Electric Network Solutions the Matrix Way by Robert Ellis
Engineers! Practicle routines for using matrix algebra.
The A.M.U.G. BBS List compiled
by Joe Rothman, Chet Solace, and Dorothy Dean 100
514 BBS phone numbers in the U.S. and Canada.
FACC II reviewed by Graham Kinsey
Put a firecracker under your floppy drives.
Uninvited reviewed by Kenneth E. Schaefer
When was the last time a game scared you to death?
Flow reviewed by Pamela Rothman
Turn your brainstorms into mental works of art.
Benchmark Modula-2 Compiler reviewed by Richie Bielak
Program development that beats Pascal to the punch.
Modula-2 Programming by Steve Faiwiszewski
The gameport device and simple sprites in action.
AmigaNotes by Richard Rae
A1000! Create a software-switchable output filter.
Roomers by The Bandito
Inside AmiExpo ... Kickstart 1.4 .. Commodores 'R' Us?
The Big Picture by Warren Ring--Unwiedly arguments to system calls? Discover the Unified Field Theory!!
Plus a great collection of monthly columns...

Volume 3 Number 4 1988

Highlights from AmiExpo, Los Angeles by Steve Hull
The Amiga shows off its best in the West.
Writing a SoundScape Patch Librarian by Todor Fay
Get your hands dirty working within the System Exclusive.
Upgrade Your A1000 to A500/2000 Audio Power—by H.Bassen
Modifications to help your A1000 make sweet music, too!
Amiga Audio Guide
Descriptive listing of all Amiga audio products.
Gels in Multi-Forth by John Bushakra
Push Gels to the limit with these programming tools.
Macrobatics by Patrick J. Horgan Ease the trauma of assembly language programming.
Amiga Audio Sources The folks behind all those audio products.
Take Five! by Steve Hull
Four lightning-paced titles to slash boredom.
Amiga Notes by Rick Rae
Confounded by sound? Take a basic tour of Amiga audio.
The Ultimate Video Accesory, Part V by Larry White
Let's add some flash to our video.
Bug Bytes by John Steiner
The Big Picture by Warren Ring
Part II of the eye-opening Unified Field Theory.
Roomers by The Bandito
Hardware hijinx ... Toasted video ... the dream Amiga ... and more!
In the Public Domain by C.W. Flatte
C.W. has hooked the latest Fish disks—here's an inside look.
Time Bandit review by Keith Conforti
A whole video arcade wrapped up in one game!
AudioMaster review by Brendan Larson
Friendly digitizing software that samples in real-time.
Music Mouse review by J Henry Lowengard
Making music without lifting a finger from the mouse.
Amiga-Tax/Canadian Version review by Ed Bercovitz
A Canadian income tax planning, preparation, & analysis package.
SAM BASIC review by Bryan Catley
A new BASIC which exploits even more unique Amiga features.

Volume 3 Number 5 1988

Interactive Startup Sequence by Udo Pernisz
The Command Line part 1by Rich Falconburg
AmigaTrix III by Warren Block—Tips and tidbits to ease Amiga life
Amiga Product Guide: Hardware Edition
Proletariat Programming by P Quaid—Public domain compilers
The Companion by Paul Gosselin
The Amiga's Event Handling capability.
MindLight 7 reviewed by David N. Blank
Psychedelic fad of the 70's updated for the Amiga.
VideoScape 3-D 2.0 reviewed by David Hopkins
Extend reviewed by Bryan D. Catley—An AmigaBASIC extension
AssemPro reviewed by Stephen Kemp
Opening a door to assembly language programming.
APL.68000 reviewed by Roger Nelson
Book Reviews by Richard Grace—Three "C" programming texts.
CBTREE reviewed by Michael Listman
A tidy collection of functions to aid the C programmer.
The Big Picture by Warren Ring
The three-part Unified Field Theory winds up
Modula-2 by Steve Faiwiszewski
Termination modules for Benchmark and TDI compilers.
68000 Assembly Language by Chris Martin
Peeling away the complication of display routines.
Plus a great collection of monthly columns...

Volume 3 Number 6 1988

Bear Time Reviewed by Steve Carter
What makes this inexpensive A1000 battery-backed clock tick?
Acquisition Reviewed by David N. Blank
A look inside the latest release of a powerful relational database.
Butcher 2.0 Reviewed by Gerald Hull
A tidy collection of diverse image processing utilities.
Reassigning Workbench Disks by John Kennan
Endless disk swapping comes to a merciful end.
Product Guide: Software Tools Edition
A listing of all the products you need to put your Amiga to work.
An IFF Reader in Multi-Forth by Warren Block
Create an easy to use IFF reader in Multi-Forth.
Basic Directory Service Program by Bryan Catley
A programming alternative to the GimmeeZeroZero windows.
C Notes from the C Group by Stephen Kemp
A beginner's guide to the power of C programming.
An Amiga Forum Conference with Jim Mackraz
The Amiga market as seen by the "Stepfather of Intuition."
Son of Seven Assemblers Reviewed by Gerald Hull
A ocmparative battle between seven native-code assemblers.
The 1988 Commodore Amiga Developers Conference
A look inside the conferences held in Washington, D.C.
Amiga Working Groups by Perry Kivolowitz and Eric Lavitsky
An outline of the innovative Amiga Working Groups concept.
The Command Line by Rich Falconburg
Exploring the multi-talented LIST command.
Plus a great collection of monthly columns...

Volume 3 Number 7 1988

Look, Up On the Screen, It's an Ami... It's a Pro... It's SuperGen
reviewed by Larry White—Genlock comparisons
An Interview with "Anim Man," Gary Bonham by B. Larson
An animated conversation with the man behind the format.
The Amiga at Spring COMDEX in Atlanta by Ed Bercovitz
The Amiga and its third party partners wow the South.
Amiga Product Guide: Video/Graphics Edition
Thirteen pages devoted to the Amiga's dazzling strong suit.
The Developing Amiga by Steve Pietrowicz
Developers' notes: PD vs. shareware vs. freely distributableetc.
Roll Those Presses! by Barney Schwartz
Welcome to the dandy, demanding world of desktop publishing!
Linked Lists in C by William E. Gammill
Put dynamic memory to work with linked lists.
FrameGrabber Preview by Oran Sands
Capturing an image can now be as fast as punching a single key!
A First Look at Interchange reviewed by David Hopkins
Bridge the gap between those incompatible animation packages.
Perfect Vision reviewed by Bryan Catley
Capture, digitize and save pictures from any video source.
ProWrite 2.0 Review reviewed by Pamela Rothman
A graphic word processor specializing in efficient editing.
Doug's Math Aquarium: The Art of Mathematics by Richie Bielak
Solving equations was never this much fun!
Bear Products MegaRex II Expansion RAM by Steve Carte
An inexpensive way to pump your Amiga up to 2MB.
The Command Line by Rich Falconburg
Amiga Notes by Rick Rae
The Other Guys' Synthia: A digital synthesizer features won't stop.
C Notes from the C Group by Stephen Kemp
Weathering the unknown "C" of basic object and data types.
Plus a great collection of monthly columns...

Volume 3 Number 8 1988

The Command Line by Rich Falconburg
The journey into the CLI continues.
The Developing Amiga by Stephen R. Pietrowicz
A gaggle of great programming tools.
Modula-2 Programming by Steve Faiwiszewski
Libraries and the FFP and IEE Math Routines.
C Notes from the C Group by Stephen Kemp
Arrays and pointers unmasked.
Dark Castle reviewed by Keith Conforti—The Black Knight lurks
Ports of Call reviewed by Julie Landry
Leatherneck reviewed by Michael Creeden-Rambo's not so tough!
Capone reviewed by Joyce and Robby Hicks- Light Guns blaze
Casino Fever reviewed by Michael T. Cabral—Vegas on Amiga
Ferrari reviewed by Jeffery Scott Hall—Start your engine
Arkanoid reviewed by Graham Kinsey—"blockbuster"
Ebonstar by Keith Conforti —black hole treking.
Deluxe Productionsreviewed by Harv Laser—Video wizardry
Game Pizazz by Jeffery Scott Hall— Register your questions here.
TrackMouse by Darryl Joyce
Convert a standard Atari trackball into a peppy Amiga TrackMouse.
Amiga Interface for Blind Users reviewed by Carl W. Mann
An ingenicus interface that opens the Amiga to even more users!
Video in the Sunshine State reviewed by Stephen R. Pietrowicz
RGB Video Creations hosts a video unveiling!
Amiga Product Guide: Games Edition
Tumblin' Tots by David Ashley
An assemply language game you can program.
Plus a great collection of monthly columns...

Volume 3 Number 9 1988

The Kideo Tapes by John Dandurand
A Georgia elementary school puts desktop video to work.
Speeding Up Your System by Tony Preston
Fuel inject your system with floppy disk caching
Amiga Product Guide: Education Edition
Everything you need to send your Amiga to the head of the class.
Computer Aided Instruction by Paul Castonguay
Authoring system in AmigaBASIC.
Gels in Multi-Forth, Part II: Screenplay by John Bushakra
Make the IFF converter from Part I easy to use-gadgets, menus,etc.
AmiExpo Midwest '88 by Michael T. Cabral
After taking the coasts by storm, the Amiga wows Chicago
Intellitype by Harv Laser—Learning to type made easy...and *fun?*
Shakespeare by Barney Schwartz—Desktop publishing in full color.
XSpecs 3D by Steve Hull—A new dimension in Amiga graphics.
AmigaNotes by Richard Rae—How IFF sound samples are stored?
Take Five! by Steve Hull—Beat the back-to-school blues !
The Command Line by Rich Falconburg—continuing tour of CLI.
Hot on the Shelves by Michael T. Cabral & Michael Creeden
What do you get when you combine intense war strategy with a monochrome monitor and desktop presentation? Check it out.
Bug Bytes by John Steiner
C Notes from the C Group by Stephen Kemp
Operators, expressions, and statements in C uncovered.
Roomers by The Bandito
Can an Apple IIgs Plus a day keep the Amiga away?

To be continued.........

To Order Back Issues,
please use the order form on page 112

The Developing *AMIGA*

by Stephen R. Pietrowicz

Beta Testing

"Oh, that product is still in beta testing. It won't be available for another two weeks." How many times have you heard *that* before? Two weeks seems to be the amount of time everyone gives for of a product's release date. Well, what exactly is "beta testing" anyway?

Beta testing is a period in the product development cycle when the product is sent out to independent testers. The testers report bugs to the developer and offer suggestions about the product. The developer fixes the bugs, and sometimes incorporates the tester's suggestions. At the end of testing, the testers are either paid, or given a release copy of the new software.

It's important that developers have *independent* beta testers look at the product, because a developer may not try everything an ordinary user tries. Ask any software developer that has gotten bug reports from customers. People always come up with ways to use programs that the developer hadn't even thought of.

Since independent testers haven't followed the product through the development cycle, they have no preconceived notions of how the product should work. The developer can gain substanially from this arrangement by improving the product before it ships. Some companies admit they do testing in-house. I think they're really missing an opportunity to develop better products.

How should you, the developer, pick beta testers? That's a really tricky question. After all, you don't want to give it to someone who is going to pirate the program, or to someone who just wants the program and isn't going to be much help in testing.

The best thing is to distribute the product to trustworthy friends, and only to people with a genuine interest in your product. Make discreet inquiries to people you think may be willing to help. It's disheartening to send a product to a beta tester, and then to hear little or no response. Be careful who you pick. Other developers may suggest reliable testers and testers you should avoid.

Once you've chosen the testers, send each a beta test contract. The contract outlines what is expected of testers, and also helps protect yourself. The testers should have a clear understanding of what is expected: how to report bugs, how testers will be paid, the terms for breaking the agreement, etc.

Once you've set up a relationship with testers, you might want to set up a BBS system to keep in close contact with them. A BBS can serve as a focal point where bug reports are submitted, new beta software is downloaded, and discussions about the current beta can take place. Services such as People/Link can also be used this way. (In fact, that's how I keep contact with developers when I beta test products.) You may want to give your phone number to your testers, so they can call with questions. Some problems are explained more easily over the phone.

Listen to your testers! If one or more of them is having a problem understanding how something works, chances are that future customers will have problems, too. Weigh each suggestion carefully. Is the suggestion something that should be added to the product now, or should you wait to add it to a future release?

The length of the beta testing period is up to the developer. It should be at least a month or longer. You're going to spend quite a bit of money for diskette duplication and documentation, so it's in your best interest to test that software throughly. After all, you don't want to spend more money duplicating disks for an update to fix a serious bug testers could have found.

Taking that extra "two weeks" for beta testing saves more than time and money; it also helps you deliver a better product to your customers. It takes a long time to establish a good reputation for your company and your products. It takes only one product to do damage to the reputation you work so hard to maintain.

USENET Software Update

In my last column, I promised to let you know when UUCP and USENET news became available for the Amiga. Dan Schein of Sneakers Computing posted a notice to the USENET, and I'd like to paraphrase it for you:

The AmigaUUCP software ported from the GNU project UUCP by William Loftus, and the Amiga news software, was ported by Dan Schein. They are now making this software available. To obtain a copy of UUCP/UseNet, send 2 blank, 3.5" disks, along with a prepaid, and self addressed return disk mailer to:

Sneakers Computing
2455 McKinley Ave.
West Lawn, PA 19609

Or if you prefer, send $10 and your name/address to the above address to cover disks and shipping costs.

One disk contains the binaries for AmigaUUCP and the news software; the other disk contains the Lattice source.

If you haven't used UUCP before, get advice from someone who has. Documentation on how to set up UUCP is not included with the software.

Two books are recommended to help you learn about UUCP:

Managing UUCP and Usenet
Tim O'Reilly and Grace Todino
O'Reilly & Associates, Inc.
1-800-338-NUTS (Cost is about $17.00)

UNIX Communications
The Waite Group
Howard W. Sams & Company.
(Cost is about $26.95)

If you have questions, you can contact Dan Schein through Berk's BBS: (215) 678-7691, 3/12/2400 Baud, 24 hrs.

New Products

Commodore recently announced the availability of the following products to dealers. Thanks to Andre Frech for this information:

A2058 8MB RAM Expansion for the A2000. Comes populated with 2M and sockets for the other 6M, expanding the A2000 to full RAM capacity. These are being substituted for the A2052 2M RAM card which is no longer in stock. Price is around $800.00.

A2090A A2000 Hard Disk Controller with Autoboot. (Just appeared on our ordering computers today... I think.) Same as the A2090, except for 1.3 autoboot capability and some rough areas that have been fixed. Price is about $399.95.

1084S Universal RGB monitor with Stereo. Same as the 1084, except it has two speakers and two input jacks. Approximate price is $420.00.

Wrap Up

What type of information would you like to see in this column? If you have any news, suggestions, questions, or programming hints, please feel free to send them to:

The Developing Amiga
c/o Amazing Computing
PO Box 869
Fall River, MA 02722-0869

I can also be reached through People/Link. Send E-mail to CBM*STEVE.

Stephen R. Pietrowicz is an assistant chairman of People/Link's Amiga Zone, a freelance writer, author of ACO—the sound and graphics conferencing program, and a member of The C Group.

•AC•

Index of Advertisers

Advertiser	Page	Reader Service Number
Amazing Computer Systems	33	133
Celestial Systems	73	173
Central Coast Software	5	105
Computer Mart	29	129
Computer Outlet	70	170
D-Five Associates	79	179
Delphi Noetic Systems	35	135
Elipsys	80	180
Erich Stein & Associates, Inc.	103	203
Expansion Technologies	91	191
EZ-Soft	99	199
Fisher's Computers & Software	73	273
Flexible Data Systems	77	177
Graphic Expressions	40	140
HyperTek/Silicon Springs	51	151
Interactive Video Systems	27	127
Lattice	9	109
Lynn's Luna C	43	143
Megatronics	CII	271
Metran Technology	68	168
Metropolitan Computer Products	24	124
Micro Way	60	160
Micro-Systems Software	47	147
MicroBotics, Inc.	39	139
Microsmiths, Inc.	36	136
New Horizons Software	7	107
New Tek	CIV	275
New Wave Software	18	118
One Byte	66	166
OTG Software	44	144
Palomax	13	113
Peacock Systems, Inc.	53	153
Pioneer Computing	69	169
Practical Solutions	101	201
Rainbird	1	102
Ronin Research & Development Inc.	4	104
Sedona Software	55	155
Soft-Link	37	137
Software Advantage Consulting Corp.	74	174
Software Sensations	10	110
Software Terminal	14	114
Software Visions	25	125
The Computer Club	72	172
The Memory Location	32	132
The Other Guys	11	111
The Sterling Connection	62	162
Tru-Image	73	274
TV One Production	76	176

To use the Reader Service Card, simply locate all the Amazing Advertisers you wish to contact, find the corresponding AC Reader Service Card Numbers, and mark them on the card to the right. Fill in the address and mail the card.

BASIC Linker

by Brian Zupke

It was 4:27 a.m. My eyes were bloodshot from staring at the unforgiving computer screen. After the 13th major modification to my latest BASIC program, it still wouldn't work. I was running out of line numbers. It took forever to trace all the GOTO's. I wondered if I had used a variable name in one of the new subroutines that already existed in the program. The pages of my reference manual were worn from verifying the memory locations I was POKEing around in. This is how I spent every night. I lost my job and my wife, and even my dog left me. My life was a complete mess. That is, until I found AB.

My friends (the few I had left) banded together and introduced me to the Amiga and AmigaBASIC. I picked up the AmigaBASIC manual, flipped through it, and saw the light. Line numbers weren't necessary, GOTO's were rarely needed, subprograms had their own local variables, and the language was so powerful that there was no need for me to POKE around. I was saved! No longer did I have to stay up all night debugging programs. I got a new job, my wife returned, and even my dog came back. (He ate my old computer though!) My life was renewed.

I thought I was okay. There I was, creating structured programs, creating subprograms, not using GOTO's. I thought I had it all. Not so. I discovered other languages created executable files from programs and subroutines automatically. I was spending lots of time MERGEing the latest versions of my subprograms and cursing myself when I got the UNDEFINED SUBPROGRAM error because I'd forgotten to MERGE one of them.

I also found out that there are four fundamental program structures, of which AmigaBASIC has only two: the IF/THEN/ELSE block and the WHILE loop. Missing is the SWITCH/CASE or, dare I say, computed GOTO (aaaacck!!) structure and the REPEAT/UNTIL loop. The REPEAT/UNTIL loop is similar to the WHILE loop except that a REPEAT/UNTIL will be executed at least once and a WHILE loop may not be executed at all. I feared I would lose control again, but I came to my senses: I had AB to save the day.

I decided not to live without these missing features, so I wrote a BASIC Linker program called BLINK. This program combines the individual routines from my program library to create an executable program. At the same time, it performs REPEAT/UNTIL and SWITCH/CASE macro expansions. That is, it converts those structures into statements AmigaBASIC can understand. I was in control again.

For those of you who are interested in adding these capabilities to your AmigaBASIC, read on.

In order to use BLINK properly, the rules and naming conventions below must be followed. They are necessary to guarantee that your programs will be linked and macros expanded correctly.

Subprogram Calls:

1) All routines (programs and subprograms) must be stored in ASCII format. This is necessary for BLINK to be able to read and combine routines correctly. Although more disk space is used to store each routine, you can use any text editor that will save files in ASCII format. If you use the AmigaBASIC editor, save your files with a comma and the letter A (",A") following their names so they will be saved in ASCII format.

2) All routines must be located on the same disk or directory. This collection of routines is referred to as the program or source library. The program library is used as the input for BLINK.

3) All calls to subprograms must use the CALL statement. Although AmigaBASIC allows a subprogram to be called by specifying only its name, BLINK uses the CALL statement to distinguish between a call statement and a statement label.

4) All subprogram names must end with ".MSB". This simplifies BLINK's task of identifying which CALL statements are subprogram calls and which are runtime library or assembly language calls. Note that the main program does not need this extension since no routine will CALL it.

5) The subprogram name and its file name must be identical. If they aren't, BLINK will not be able to find the subprogram.

6) Only one subprogram call per line is permitted. If there is more than one CALL on a line, BLINK will see only the first one.

REPEAT/UNTIL usage:

The sequence:

```
REPEAT
  .     \
  .  >-  Valid statements
  .     /
UNTIL(condition)
```

is expanded by BLINK to:

```
RU(x) = -1
WHILE RU(x)
  .
  .
  .
IF (condition) THEN RU(x) = 0
WEND
```

where:

x -nested level of the REPEAT/UNTIL loop. This will actually be a number from 0 to 9. No variable x will be used.

1) The REPEAT and UNTIL statements must appear on lines by themselves.

2) The condition portion of the UNTIL statement must be a valid BASIC expression that will yield a TRUE/FALSE result when evaluated. This expression must be enclosed by parenthesis, and spaces are not permitted between "UNTIL" and "(".

SWITCH/CASE usage:

The sequence:

```
             SWITCH(expression)
                CASE value1
                    .
       .
                CASE value2
           .
       .
   CASE value-n
       .
       .
   DEFAULT
       .
       .
SWEND

    is expanded by BLINK to:

             IF expression = value1 THEN
    .
    .
ELSEIF expression = value2 THEN
    .
    .
ELSEIF expression = value-n THEN
    .
    .
ELSE
    .
    .
             END IF
```

1) All SWITCH, CASE, DEFAULT, and SWEND statements must appear on separate lines.

2) For each SWITCH statement there must be at least one CASE statement and only one SWEND statement. The DEFAULT statement is optional. If no DEFAULT statement is used, then the last ELSE block will not appear.

3) The expression portion of the SWITCH statement must be a valid AmigaBASIC expression enclosed in parenthesis. Spaces are not permitted between "SWITCH" and "(".

4) There is no limit to the number of CASE statements that can be in a SWITCH structure.

5) The value portion of the CASE statement must be a valid AmigaBASIC expression ending with a colon ":". It must be compatible with the expression in the SWITCH statement.

6) SWITCH/CASE structures may be nested in up to nine levels.

TRUE/FALSE conversion:

When this function is enabled in BLINK, all occurrences of upper-case only "TRUE" and "FALSE" will be converted to -1 and 0, respectively. If you link BLINK with this conversion active, the main program's statements:

```
TRUE = -1          FALSE = 0
```

will be converted to:

```
-1 = -1          0 = 0
```

which will cause a syntax error. Therefore, do not link BLINK with this conversion active. The TRUE/FALSE macro conversion serves more as an example of what kind of conversions are possible. I prefer to have the variables TRUE and FALSE DIMensioned as SHARED in the main program as they are in BLINK.

Entering the program:

Since you can't link BASIC programs until you've created BLINK (a small Catch-22), enter all of BLINK's subprograms as a single source. Use the test programs to verify that it is working properly (see instructions below on using BLINK). Once the program is working, save each subprogram under its proper name in your program library. BLINK should be able to link itself now.

Using the program:

When you want to link a program, run BLINK from AmigaBASIC and enter the information requested by the program. You will be asked for the following items:

Program name: the name of the program to be linked.

Library pathname: the disk or directory containing all of your AmigaBASIC programs and subprograms.

Program pathname: the disk or directory in which BLINK will place the linked program. The linked program will be saved under the same name, with ".EXE" added to the end of it indicating that it is executable.

Compress (Y or N): all comment lines will be removed from the linked program if requested. Only lines that have "'" or "rem" or "REM" in the first column will be removed. This feature is useful for cutting down on the size and increasing the speed of a finished program. When BLINK itself is linked with no comments, its speed is approximately doubled.

TRUE/FALSE Conversion (Y or N): all occurrences of upper case "TRUE" and "FALSE" will be replaced with " -1 " and " 0 " respectively.

Once this information is entered, the program begins reading the specified program from the source library and storing it under its new name. During this transfer, BLINK searches each line of code for AmigaBASIC calls and adds the names of any it finds to a list of routines that will also be transferred. After the main program has been processed, BLINK will begin transferring the next subprogram on the list. BLINK will also work on a program that has no subprogram calls. BLINK prints the name of the source it is currently scanning and linking. Once all the routines are transferred, the name of the new file is printed. If BLINK cannot find a subprogram in the library, it will report the error and stop executing.

Debugging your program:

When you're testing your newly created programs, be sure to make the final corrections to the routines in your library rather than in the linked version.

Enhancing/modifying BLINK:

If you wish to add other macro-expansion/conversion routines to BLINK, add them to the routine ScanModule.MSB. There are two sections of macro expansion calls in this routine. The first consists of macro conversion routines that modify only part of a BASIC line (such as TRUE/FALSE). A single BASIC source line can contain more than one conversion macro call. (Note that the line may be altered by more than one conversion routine.) The other section contains macro expansion routines whose expansions take up more than one line. Only one of these can exist per line of original BASIC code. To add a macro expansion here, simply follow the "Future Macro" skeleton.

Conclusion:

I kept the routines in BLINK as simple as possible, so they may not handle every possible condition. Since the code is somewhat simple, it should be easy to customize. Enjoy!

LISTING ONE

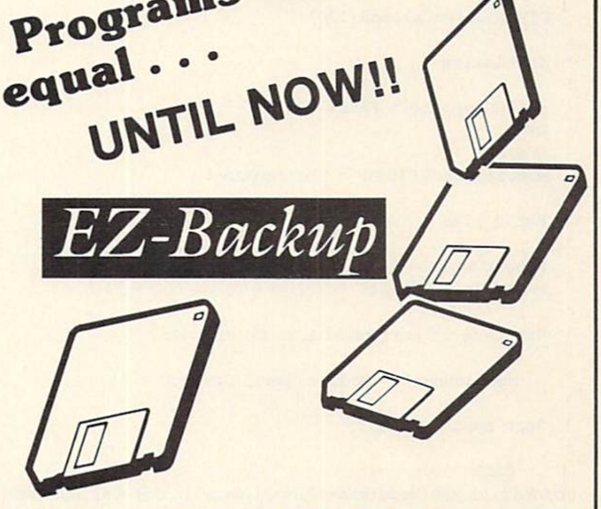
```
SUB GetLinkerInfo.MSB(ProgramName$,LibPath$,Prgpath$) STATIC
'
'  Copyright 1988 by Brian Zupke
'
'
'     Gets the program name to be linked, library and
'   executable program pathnames, compress and TRUE/FALSE
'   conversion commands.
'
'
   SHARED Compress,TFConvert
'
'  Make sure program name is valid
'
   ProgramName$ = ""
   WHILE ProgramName$ = ""
      INPUT "Name of program to be linked";ProgramName$
   WEND
'
'  Get library pathname
'
   INPUT "Pathname of program library";LibPath$
   IF LEN(LibPath$) > 0 AND RIGHT$(LibPath$,1) <> "/" AND
RIGHT$(LibPath$,1) <> ":" THEN LibPath$ = LibPath$ + "/"
'
'  Get destination pathname
'
   INPUT "Pathname (destination) of linked program";PrgPath$
   IF LEN(PrgPath$) > 0 AND RIGHT$(PrgPath$,1) <> "/" AND
RIGHT$(PrgPath$,1) <> ":" THEN PrgPath$ = PrgPath$ + "/"
'
'  Set Compression flag
'
   INPUT "Compress program (Y or N)";Answer$
   IF UCASE$(Answer$) = "Y" THEN
      Compress = TRUE
   ELSE
      Compress = FALSE
   END IF
'
'  Set TRUE/FALSE conversion flag
'
   INPUT "TRUE/FALSE Conversion (Y or N)";Answer$
   IF UCASE$(Answer$) = "Y" THEN
      TFConvert = TRUE
   ELSE
      TFConvert = FALSE
   END IF
END SUB
SUB
LinkModules.MSB(ProgramName$,ModulesUsed$(1),Used,BuildComplete)
STATIC
'
```

BREAKTHROUGH

All Hard Drive Back-Up Programs were created equal . . .

UNTIL NOW!!

EZ-Backup



100% better than any other hard drive back-up program
EZ-Backup is a genuine breakthrough

EZ-Backup actually manages the space on your backup disks--

Before EZ-Backup the only way to get rid of the piles of incremental back-up disks that littered your desk was to do another full back up. This method is inconvenient and requires a whole new set of unformatted floppy disks.

With EZ-Backup you use the same set of disks for every incremental back-up. Only one full back-up required--Ever! Space on the disks is managed by deleting obsolete archive files and allowing you to save from 0-255 versions of each file. **Your files are saved--even if you have completely deleted them from the hard drive!**

EZ-Backup comes with an optional warning screen--
We all tend to put off doing backups. EZ-Backup's warning screen reminds you. If you would rather not be reminded--you have the option to shut the warning screen off.

EZ-Backup prevents you from damaging valuable data--
By checking the volume label, EZ-Backup keeps you from writing over important files.

EZ-Backup uses Standard Amiga format--
Files are archived in standard Amiga format and work with all standard utilities.

EZ-Backup provides easy recovery of individual files--
A simple to use mouse-oriented program allows you to recover individual files.

Works with all Amiga-DOS compatible hard drives.
(Amiga-DOS version 1.2 or higher)

- ⋆ Provides archive-bit utilities
- ⋆ Not copy protected
- ⋆ Multi-tasking
- ⋆ Complete manual with examples
- ⋆ Upgrades provided free for the first six months after program purchase!
- ⋆ Free telephone support!

Not more expensive just the best - $49.95 - from:
EZ-SOFT or an Amiga Dealer near you.
21125 Chatsworth Street
Chatsworth, CA 91311
(818) 341-8681
Dealer Inquires Welcome

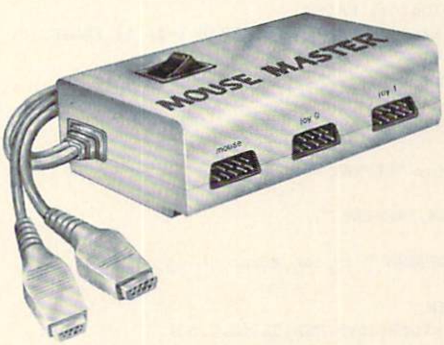
EZ SOFT

```
' Copyright 1988 by Brian Zupke
'
' Links all modules used in the main prog to create executable.
' If all modules are found, then BuildComplete is flagged TRUE.
'
'
' Allow up to 100 different Subprograms.
'
  DIM ModulesCalled$(100)
'
' Initialize
'
  BuildComplete = FALSE
  Used = 1
  Scanned = 0
  ModulesUsed$(Used) = ProgramName$
'
' Build List
'
  RepeatUntil = TRUE
  WHILE RepeatUntil
'
' Get name of next module to be scanned.
'
     ModuleName$ = ModulesUsed$(Scanned + 1)
'
' Scan next module
'
     CALL
ScanModule.MSB(ModuleName$,ModulesCalled$(),Called,ModuleScanned)
'
' Compare any calls found with modules used list
'
     IF Called > 0 THEN
        FOR x = 1 TO Called
           IsNotIn = TRUE
           FOR y = 1 TO Used
              IF UCASE$(ModulesCalled$(x)) =
                 UCASE$(ModulesUsed$(y)) THEN IsNotIn = FALSE
           NEXT y
'
' Add name to list if it is not already there
'
           IF IsNotIn THEN
              Used = Used + 1
              ModulesUsed$(Used) = ModulesCalled$(x)
           END IF
        NEXT x
     END IF
     Scanned = Scanned + 1
     IF (Used = Scanned) OR (NOT ModuleScanned)
        THEN RepeatUntil = FALSE
  WEND
'
' Build is complete if last module scanned successfully
'
  BuildComplete = ModuleScanned
END SUB
SUB
ScanModule.MSB(ModuleName$,ModulesCalled$(1),Called,ModuleScanned)
STATIC
'
' Copyright 1988 by Brian Zupke
'
'
'    Scans module using LibraryPathname$ for calls to subpro-
grams
' in the user library (those having the '.MSB' extension).
Names
' of subprograms found are placed in ModulesCalled$ list.  If
' the module scan fails, then ModuleScanned is returned FALSE.
' Each line read is also place in destination file to create
' the executeable program.  Macro expansions and conversionsare
' also performed as lines are transferred to destination file.
'
'
  SHARED LibraryPathname$,Compress,TFConvert
'
```

```
  Called = 0
  PRINT "Scanning & Linking:  ";ModuleName$
'
  OPEN LibraryPathname$+ModuleName$ FOR INPUT AS #1 LEN = 4096
'
' Read each line of the current module from library.
'
  IF NoError THEN
     WHILE NoError AND (NOT EOF(1))
        LINE INPUT#1,SourceLine$
'
' line is to be output and no expansion occurred for now
'
        OutputLine = TRUE
        Expanded = FALSE
'
' Check if current line is comment
'
        CALL CommentCheck.MSB(SourceLine$,LineIsComment)
        IF LineIsComment THEN
           IF Compress THEN OutputLine = FALSE
        ELSE
'
' Check if current line has subprogram call
'
           CALL FindCall.MSB(SourceLine$,ModuleCall$)
           IF ModuleCall$ <> "" THEN
              Called = Called + 1
              ModulesCalled$(Called) = ModuleCall$
           END IF
'
' ____________________________
' Macro conversions that change only part of the line go here:
' ____________________________
'
' Expand TRUE/FALSE macro calls if requested
'
           IF TFConvert THEN CALL
                 TrueFalseMacro.MSB(SourceLine$)
'
' ____________________________
' Macro expansions that change entire line go here:
' ____________________________
'
' Expand REPEAT/UNTIL macro calls
'
           IF (NOT Expanded) THEN
              CALL RepeatUntilMacro.MSB(SourceLine$,Expanded)
           END IF
'
' Expand SWITCH macro
'
           IF (NOT Expanded) THEN
              CALL SwitchMacro.MSB(SourceLine$,Expanded)
           END IF
'
' Future macro
'
'          IF (NOT Expanded) THEN
'             CALL would go here
'          END IF
'
' No need to output to destination file if entire line changed
'
           IF (Expanded) THEN OutputLine = FALSE
        END IF
'
' Place current line in output file as required
'
        IF OutputLine THEN
           PRINT #2,SourceLine$
        END IF
     WEND
  END IF
'
' Report results
'
  IF NoError THEN
```

```
      ModuleScanned = TRUE
   ELSE
      ModuleScanned = FALSE
   END IF
   CLOSE #1
END SUB
SUB CommentCheck.MSB(SourceLine$,LineIsComment) STATIC
`
`
` Copyright 1988 by Brian Zupke
`
`
`     Scans source line in the form of a character string
` and checks if line is a comment.  A comment line will be
` blank or the first characters will be "`" or "rem".
`
   LineIsComment = FALSE
   SL$ = UCASE$(SourceLine$)
`
   IF LEN(SL$) = 0 OR LEFT$(SL$,1) = "`" OR
         LEFT$(SL$,3) = "REM" THEN
      LineIsComment = TRUE
   END IF
END SUB
SUB FindCall.MSB(SourceLine$,ModuleCall$) STATIC
`
`
` Copyright 1988 by Brian Zupke
`
`
`     Scans source line for `CALL X.MSB' where X is a
` module name.  If found, then the module name is
` returned in ModuleCall$.
`
`
   ModuleCall$ = ""
   SL$ = " " + SourceLine$
`
` Locate ` CALL `
`
   C = INSTR(UCASE$(SL$)," CALL ")
   IF C > 0 THEN
`
` Isolate subprogram name
`
      Character = C + 6
      WHILE MID$(SL$,Character,1) = " "
         Character = Character + 1
      WEND
      C = Character
`
` Check for ".MSB" extension - if there, then module call found
`
      C = INSTR(C,UCASE$(SL$),".MSB")
      IF C > 0 THEN
         Length = (C + 4) - Character
         ModuleCall$ = MID$(SL$,Character,Length)
      END IF
   END IF
END SUB
SUB TrueFalseMacro.MSB(SourceLine$) STATIC
`
`
` Copyright 1988 by Brian Zupke
`
`
` Scans SourceLine$ for the occurance of "TRUE" or "FALSE" and
`  replaces them with " -1 " or "  0  ".
`
`
` Initialize if first time in routine.
`
   IF NOT Initialized THEN
      Label$(1) = "TRUE"
      Label$(2) = "FALSE"
      Value$(1) = " -1 "
      Value$(2) = "  0  "
      Initialized = TRUE
   END IF
`  Search for both TRUE and FALSE
`
   FOR x = 1 TO 2
      C = 1
      WHILE C > 0 AND C < LEN(SourceLine$)
         C = INSTR(C,SourceLine$,Label$(x))
         IF C > 0 THEN
            MID$(SourceLine$,C,3+x) = Value$(x)
            C = C + 4 + x
         END IF
      WEND
   NEXT x
END SUB
SUB RepeatUntilMacro.MSB(SourceLine$,Expanded) STATIC
`
`  Copyright 1988 by Brian Zupke
`
`
`     Checks string SourceLine$ for "REPEAT" or "UNTIL(".
`  If found, writes the expansion to the output file #2 and sets
`  the Expanded flag to TRUE.
`
`  the structure:
`
`     REPEAT
`        .
`        .
`     UNTIL (condition)
`
`  expands to:
`
`     RU(x) = -1
`     WHILE RU(x)
`        .
`        .
`     IF (condition) THEN RU(x) = 0
`     WEND
`
```

Now For The Amiga!



Are you tired of fumbling under or behind your computer to swap your mouse and joystick cables? Are your cable and computer connectors worn out from all the plugging and unplugging? Then Mouse Master is a must for you!

$39.95*

Practical Solutions®

1930 E. Grant Rd.,
Tucson, AZ 85719

602-884-9612

*Retail price does not include shipping & handling.

MOUSE MASTER

```
' where:
'
'        x - nested level of the REPEAT/UNTIL loop (0-9)
'       -1 - logical TRUE
'        0 - logical FALSE
' condition - expression that is TRUE (-1) or FALSE (0)
'            when evaluated
'        . - one or more Amiga BASIC statements
'
  SL$ = " " + UCASE$(SourceLine$) + " "
'
' Check line for "REPEAT "
'
  C = INSTR(SL$,"REPEAT ")
'
' Make sure "REPEAT " is the first thing on line.
'
  IF C > 0 THEN
    CALL CheckForBlanks.MSB(SL$,C,Good)
'
' Expand "REPEAT " to output file #2
'
     IF Good THEN
        PRINT #2,SPC(C-2);"RU(";RepLvl;") = -1"
        PRINT #2,SPC(C-2);"WHILE RU(";RepLvl;")"
        RepLvl = RepLvl + 1
        Expanded = TRUE
     END IF
'
' Check line for "UNTIL("
'
  ELSE
     C = INSTR(SL$,"UNTIL(")
'
' Make sure "UNTIL(....)" is first thing on line
'
     IF C > 0 THEN
        CALL CheckForBlanks.MSB(SL$,C,Good)
'
' Expand "UNTIL(....)" to output file #2
'
        IF Good THEN
           RepLvl = RepLvl - 1
           PRINT #2,SPC(C-2);"IF ";MID$(SL$,C+5);"
                 THEN RU(";RepLvl;") = 0"
           PRINT #2,SPC(C-2);"WEND"
           Expanded = TRUE
        END IF
     END IF
  END IF
END SUB
SUB SwitchMacro.MSB(SourceLine$,Expanded) STATIC
'
' Copyright 1988 by Brian Zupke
'
'
' Checks string SourceLine$ for "SWITCH", "CASE", "DEFAULT", or
'  "SWEND".  If found, writes the exp to the output file #2 and
'  sets the Expanded flag to TRUE.
'
'
' the structure:
'
'   SWITCH(condition)
'      CASE value1:
'        .
'      CASE value2:
'        .
'      CASE valueN:
'        .
'      DEFAULT
'        .
'   SWEND
'
' expands to:
'
'    IF condition = value1 THEN
'        .
'    ELSE IF condition = value2 THEN
'        .
'    ELSE IF condition = valueN THEN
'        .
'    ELSE
'        .
'    ENDIF
'
' where:
'
'   condition - a valid BASIC expression
'
'   valueX - a valid BASIC expression of same type as
condition
'
'   . - one or more Amiga BASIC statements (SWITCH is permit-
ted)
'
'   DEFAULT condition is optional
'
  SL$ = " " + UCASE$(SourceLine$) + " "
'
' Check line for "SWITCH"
'
  C = INSTR(SL$,"SWITCH(")
'
' Make sure "SWITCH(" is first thing on line.
'
  IF C > 0 THEN
     CALL CheckForBlanks.MSB(SL$,C,Good)
'
' Extract condition expression
'
     IF Good THEN
        Index = C + 7
        Nparen = 1
        WHILE Nparen > 0 AND Index <= LEN(SL$)
           IF MID$(SL$,Index,1) = "(" THEN Nparen = Nparen + 1
           IF MID$(SL$,Index,1) = ")" THEN Nparen = Nparen - 1
           Index = Index + 1
        WEND
        SwitchLevel = SwitchLevel + 1
        CaseExpanded(SwitchLevel) = FALSE
        Condition$(SwitchLevel) = MID$(SL$,C+6,Index-(C+6))
'
' No output on "SWITCH" expansion
'
        Expanded = TRUE
     END IF
'
' Check for "CASE", "DEFAULT", and "SWEND" only after "SWITCH"
found
'
  ELSEIF SwitchLevel > 0 THEN
'
' Check line for "CASE "
'
     C = INSTR(SL$,"CASE ")
'
' Make sure "CASE " is first thing on line.
'
     IF C > 0 THEN
        CALL CheckForBlanks.MSB(SL$,C,Good)
'
' Extract value
'
        IF Good THEN
           LastCase = C
           Index = C + 5
          WHILE MID$(SL$,Index,1) <> ":" AND Index <= LEN(SL$)
              Index = Index + 1
           WEND
          IF CaseExpanded(SwitchLevel) THEN
              PRINT #2,SPC(C-2);
        "ELSEIF                ";
        Condition$(SwitchLevel);" = ";
        MID$(SL$,C+5,Index-(C+5));" THEN"
           ELSE
              PRINT #2,SPC(C-2);"IF ";
        Condition$(SwitchLevel);" = ";
        MID$(SL$,C+5,Index-(C+5));" THEN"
           END IF
```

```
            CaseExpanded(SwitchLevel) = TRUE
            Expanded = TRUE
         END IF
      END IF
'
' Check for "DEFAULT "
'
      IF NOT Expanded THEN
         C = INSTR(SL$,"DEFAULT ")
'
' Make sure " DEFAULT " is first thing on line.
'
         IF C > 0 THEN
           CALL CheckForBlanks.MSB(SL$,C,Good)
'
' Output "ELSE" to device #2
'
            IF Good THEN
               PRINT #2,SPC(C-2);"ELSE"
               Expanded = TRUE
            END IF
         END IF
      END IF
'
' Check for "SWEND "
'
      IF NOT Expanded THEN
         C = INSTR(SL$,"SWEND ")
'
' Make sure "SWEND " is first thing on line.
'
         IF C > 0 THEN
           CALL CheckForBlanks.MSB(SL$,C,Good)
'
' Output "END IF" to device #2
'
            IF Good THEN
               SwitchLevel = SwitchLevel - 1
               PRINT #2,SPC(LastCase-2);"END IF"
               Expanded = TRUE
            END IF
         END IF
      END IF
   END IF
END SUB
SUB CheckForBlanks.MSB(SL$,C,Good) STATIC
'
' Copyright 1988 by Brian Zupke
'
'
'    Checks string SL$ for blanks up to but not including C'th
'char.  Good is returned TRUE if no other characters are found.
'
   Good = TRUE
   FOR x = 1 TO C-1
      IF MID$(SL$,x,1) <> " " THEN Good = FALSE
   NEXT x
END SUB
```

Test. LISTING

```
' Inventory
'
   DIM SHARED F$,D$,Q$,R$,P$
'
   OPEN"RAM:INVEN.DAT" AS #1 LEN=39
   FIELD #1,1 AS F$, 30 AS D$, 2 AS Q$, 2 AS R$, 4 AS P$
'
   Done = FALSE
   REPEAT
      CLS:PRINT"Functions:":PRINT
      PRINT "1. Initialize file"
      PRINT "2. Create a new entry"
      PRINT "3. Display inventory for one part"
      PRINT "4. Add to stock"
      PRINT "5. Subtract from stock"
      PRINT "6. Display all items below reorder level"
      PRINT "7. Done with this program"
      PRINT:PRINT:INPUT "Function";FUNCT
      SWITCH(FUNCT)
         CASE 1:
            CALL InitFile.MSB
         CASE 2:
            CALL CreateEntry.MSB
         CASE 3:
            CALL ShowEntry.MSB
         CASE 4:
            CALL AddToStock.MSB
         CASE 5:
            CALL SubFromStock.MSB
         CASE 6:
            CALL ShowReorder.MSB
         CASE 7:
            Done = TRUE
      SWEND
   UNTIL(Done)
   CLOSE #1
END
SUB InitFile.MSB STATIC
'
' Initialize inventory file
'
   INPUT"Are you sure";confirm$
   IF UCASE$(confirm$) = "Y" THEN
      LSET F$ = CHR$(255)
      FOR I = 1 TO 100
         PUT #1,I
      NEXT I
   END IF
END SUB
SUB CreateEntry.MSB STATIC
'
' Create new entry in inventory file
'
```

Erich Stein & Associates, Inc.

Public Relations Consultants

Because the quality of your reputation is just as important as the quality of your product.

PO Box 695
Denver, Colorado 80201
TEL (303) 733-3707

```
   CALL GetPart.MSB(part%)
   GoAhead = TRUE
   IF ASC(F$) <> 255 THEN
      INPUT "Overwrite";confirm$
      IF UCASE$(confirm$) <> "Y" THEN GoAhead = FALSE
   END IF
   IF GoAhead THEN
      LSET F$ = CHR$(0)
      INPUT "Description ";description$
      LSET D$ = description$
      INPUT "Quantity in stock";quantity%
      LSET Q$ = MKI$(quantity%)
      INPUT "Reorder level";reorder%
      LSET R$ = MKI$(reorder%)
      INPUT "Unit price";price
      LSET P$ = MKS$(price)
      PUT #1,part%
      INPUT "Press return to continue";DUM$
   END IF
END SUB
SUB ShowEntry.MSB STATIC
'
' Display inventory for one part
'
   CALL GetPart.MSB(part%)
   IF ASC(F$) = 255 THEN
      CALL NullEntry.MSB
   ELSE
      PRINT USING "Part Number ###";part%
      PRINT D$
      PRINT USING "Quantity on hand #####";CVI(Q$)
      PRINT USING "Reorder level ####";CVI(R$)
      PRINT USING "Unit Price $$##.##";CVS(P$)
      INPUT "Press RETURN to continue";DUM$
   END IF
END SUB
SUB AddToStock.MSB STATIC
'
' add to stock
'
   CALL GetPart.MSB(part%)
   IF ASC(F$) = 255 THEN
      CALL NullEntry.MSB
   ELSE
      PRINT D$
      PRINT "Current quantity: ";CVI(Q$)
      INPUT "Quantity to add";additional%
      Q% = CVI(Q$) + additional%
      LSET Q$ = MKI$(Q%)
      PUT #1,part%
   END IF
END SUB
SUB SubFromStock.MSB STATIC
'
' Subtract from stock
'
   CALL GetPart.MSB(part%)
   IF ASC(F$) THEN
      CALL NullEntry.MSB
   ELSE
      PRINT D$
      REPEAT
         INPUT "Quantity to subtract";less%
         Q% = CVI(Q$)
         IF (Q%-less%) < 0 THEN PRINT "Only ";Q%;" in stock"
      UNTIL((Q%-less%) >= 0)
      Q% = Q% - less%
      IF Q% < CVI(R$) THEN PRINT "Quantity now ";Q%
      LSET Q$ = MKI$(Q%)
      PUT #1,part%
      INPUT "Press RETURN to continue";DUM$
   END IF
END SUB
SUB ShowReorder.MSB STATIC
'
' display all items below reorder number
'
   reorder = FALSE
   FOR I = 1 TO 100
      GET #1,I
      IF ASC(F$) <> 255 THEN
         IF CVI(Q$) < CVI(R$) THEN
            PRINT D$;" Quantity ";CVI(Q$);TAB(30)
            PRINT "Reorder level ";CVI(R$)
            reorder = TRUE
         END IF
      END IF
   NEXT I
   IF reorder = FALSE THEN PRINT "All items well-stocked."
   INPUT "Press RETURN to continue";DUM$
END SUB
SUB GetPart.MSB(part%) STATIC
'
' get a valid part number
'
   REPEAT
      INPUT "Part number";part%
      IF (part%<1) OR (part%>100) THEN PRINT "Bad part number"
   UNTIL((part%>=1) AND (part%<=100))
   GET #1,part%
END SUB
SUB NullEntry.MSB STATIC
'
' report null entry
'
   PRINT "Null Entry."
   INPUT "Please press RETURN";DUM$
END SUB
'
' BTEST1
'
   REPEAT
      INPUT"Enter a number";Number
      SWITCH(Number)
         CASE 2:
            PRINT"The number is 2
         CASE 3:
            PRINT"The number is 3
         CASE 4:
            PRINT"The number is 4 and..."
            CALL GetLetter.MSB
         CASE 9:
            PRINT "The number is 9"
         DEFAULT
            PRINT "The number is not 2, 3, 4, or 9"
      SWEND
   UNTIL(Number = 0)
SUB GetLetter.MSB STATIC
' BLINK test program
'
   GetOut = FALSE
   REPEAT
      INPUT"Enter a letter";Letter$
      Letter$ = UCASE$(LEFT$(Letter$,1))
      SWITCH(Letter$)
         CASE "A":
            PRINT "The letter is A"
         CASE "B":
            PRINT "The letter is B"
         CASE "C":
            PRINT "The letter is C"
         DEFAULT
            PRINT "The letter is not A, B, or C"
            INPUT "Enter Q to get out of GetLetter.MSB";C$
            IF UCASE$(LEFT$(C$,1)) = "Q" THEN
               GetOut = TRUE
            ELSE
               CALL PrintMessage.MSB(C$)
            END IF
      SWEND
   UNTIL(GetOut)
END SUB
SUB PrintMessage.MSB(Letter$) STATIC
' BLINK test program
   REPEAT
      PRINT "You entered: ";Letter$
      PRINT "You'll need to enter Q to get out"
      INPUT "Understand";Yes$
   UNTIL(UCASE$(LEFT$(Yes$,1)) = "Y")
END SUB
```

•AC•

The AMICUS & Fred Fish Public Domain Software Library

This software is collected from user groups and electronic bulletin boards around the nation. Each Amicus disk is nearly full, and is fully accessible from the Workbench. If source code is provided for any program, then the executable version is also present. This means that you don't need the C compiler to run these programs. An exception is granted for those programs only of use to people who own a C compiler.

The Fred Fish disk are collected by Mr. Fred Fish, a good and active friend of the Amiga.

Note: Each description line below may include something like 'S-O-E-D', which stands for 'source, object file, executable and documentation'. Any combination of these letters indicates what forms of the program are present. Basic programs are presented entirely in source code format.

AMICUS Disk 1

ABasic programs: Graphics

3DSolids	3d solids modeling prog. w/sample data files
Blocks	draws blocks
Cubes	draws cubes
Durer	draws pictures in the style of Durer
FScape	draws fractal landscapes
Hidden	3D drawing program, w/ hidden line removal
JPad	simple paint program
Optical	draw several optical illusions
PaintBox	simple paint program
Shuttle	draws the Shuttle in 3d wireframe
SpaceArt	graphics demo
Speaker	speech utility
Sphere	draws spheres
Spiral	draws color spirals
ThreeDee	3d function plots
Topography	artificial topography
Wheels	draws circle graphics
Xenos	draws fractal planet landscapes

ABasic programs: Tools

AddressBook	simple database program for addresses
CardFile	simple card file database program
Demo	multiwindow demo
KeyCodes	shows keycodes for a key you press
Menu	run many ABasic programs from a menu
MoreColors	way to get more colors on the screen at once, using aliasing
shapes	simple color shape designer SpeakIt speech and narrator demo

ABasic programs: Games

BrickOut	classic computer brick wall game
Othello	also known as 'go'
Saucer	simple shoot-em-up game
Spelling	simple talking spelling game
ToyBox	selectable graphics demo

ABasic programs: Sounds

Entertainer	plays that tune
HAL9000	pretends it's a real computer
Police	simple police siren sound
SugarPlum	plays "The Dance of the Sugarplum Fairies"

C programs:

ATerm	simple terminal program, S-E
cc	aid to compiling with Lattice C
decvnt	opposite of CONVERT for cross developers
Dotty	source code to the 'dotty window demo
echox	unix-style filename expansion, partial S,O-D
fasterfp	explains use of fast-floating point math
FixDate	fixes future dates on all files on a disk, S-E
freedraw	simple Workbench drawing prog.,S-E
GfxMem	graphic memory usage indicator, S-E
Grep	searches for a given string in a file with
docs.	ham shows off the hold-and-modify method of color generation
IBM2Amiga	fast parallel cable transfers between an IBM and an Amiga
Mandel	Mandelbrot set program, S-E
moire	patterned graphic demo, S-E
objfix	makes Lattice C object file symbols visible to Wack, S-E
quick	quick sort strings routine
raw	example sample window I/O
setlace	turns on interlace mode, S-E
sparks	qix-type graphic demo, S-E

Other executable programs:

SpeechToy	speech demonstration
WhichFont	displays all available fonts

Texts:

68020	describes 68020 speedup board from CSA
Aliases	explains uses of the ASSIGN command
Bugs	known bug list in Lattice C 3.02
CLICard	reference card for AmigaDOS CLI
CLICommands	guide to using the CLI
Commands	shorter guide to AmigaDOS CLI commands
EdCommands	guide to the ED editor
Filenames	AmigaDOS filename wildcard conventions
HalfBright	explains rare graphics chips that can do more colors
ModemPins	description of the serial port pinout
RAMdisks	tips on setting up your RAM: disk
ROMWack	tips on using ROMWack
Sounds	explanation of Instrument demo sound file format
Speed	refutation of Amiga's CPU and custom chip speed
WackCmds	tips on using Wack

AMICUS Disk 2

C programs:

alib	AmigaDOS object library manager , S-E
ar	text file archive program, S-E
fixobj	auto-chops executable files
shell	simple CLI shell, S-E
sq, usq	file compression programs, S-E
YachtC	a familiar game, S-E
Make	a simple 'make' programming utility, S-E
Emacs	an early version of the Amiga text editor, S-E-D

Assembler programs:

bsearch.asm	binary search code
qsort.asm	Unix compatible qsort() function, source and C test program
setjmp.asm	setjmp() code for Lattice 3.02
SVprintf	Unix system V compatible printf()
trees.o	Unix compatible tree() function, O-D

(This disk formerly had IFF specification files and examples. Since this spec is constantly updated, the IFF spec files have been moved to their own disk in the AMICUS collection.)

John Draper Amiga Tutorials:

Animate	describes animation algorithms
Gadgets	tutorial on gadgets
Menus	learn about Intuition menus

AMICUS Disk 3

C programs:

Xref	a C cross-reference gen., S-E
6bitcolor	extra-half-bright chip gfx demo, S-E
Chop	truncate (chop) files down to size, S-E
Cleanup	removes strange characters from text files
CR2LF	converts carriage returns to line feeds in Amiga files, S-E
Error	adds compile errors to a C file, S
Hello	window ex. from the RKM, S
Kermit	generic Kermit implementation, flakey, no terminal mode, S-E
Scales	sound demo plays scales, S-E
SkewB	Rubik cube demo in hi-res colors, S-E

AmigaBasicProgs(dir)

Automata	cellular automata simulation
CrazyEights	card game
Graph	function graphing programs
WitchingHour	a game

ABasiC programs:

Casino	games of poker, blackjack, dice, and craps
Gomoku	also known as 'othello'
Sabotage	sort of an adventure game

Executable programs:

Disassem	a 68000 disassembler, E-D
DpSlide	shows a given set of IFF pictures, E-D
Arrange	a text formatting program, E-D

Assembler programs:

Argoterm	terminal program with speech and Xmodem, S-E

AMICUS Disk 4 Files from the original Amiga Technical BBS

Note that some of these files are old, and refer to older versions of the operating system. These files came from the Sun system that served as Amiga technical support HQ for most of 1985. These files do not carry a warranty, and are for educational purposes only. Of course, that's not to say they don't work.

Complete and nearly up-to-date C source to 'image.ed', an early version of the Icon Editor. This is a little flaky, but compiles and runs.

An Intuition demo, in full C source, including files: demomenu.c, demomenu2.c, demoreq.c, getascii.c, idemo.c, idemo.guide, idemo.make, idemoall.h, nodos.c, and txwrite.c

addmem.c	add external memory to the system
bobtest.c	example of BOB use
consoleIO.c	console IO example
creaport.c	create and delete ports
creastdi.c	create standard I/O requests
creatask.c	creating task examples
diskio.c	example of track read and write
dotty.c	source to the 'dotty window' demo
dualplay.c	dual playfield example
flood.c	flood fill example
freemap.c	old version of 'freemap'
geltools.c	tools for VSprites and BOBs
gfxmem.c	graphic memory usage indicator
hello.c	window example from RKM
inputdev.c	adding an input handler to the input stream
joystik.c	reading the joystick
keybd.c	direct keyboard reading
layertes.c	layers examples
mousport.c	test mouse port
ownlib.c, ownlib.asm	example of making your own library with Lattice
paratest.c	tests parallel port commands
seritest.c	tests serial port commands
serisamp.c	example of serial port use
prinintr.c	sample printer interface code
prtbase.h	printer device definitions
regintes.c	region test program
setlace.c	source to interlace on/off program
setparallel.c	set the attributes of the parallel port
SetSerial.c	set the attributes (parity, dasertaltport the
singplay.c	single playfield example
speechtoy.c	source to narrator and phonetics demo
timedely.c	simple timer demo
timer.c	exec support timer functions
timrstuf.c	more exec support timer functions
WhichFont.c	loads and displays all available system fonts

process.i and prtbase.i assmebler include files:

autorqstr.txt	warnings of deadlocks with autorequesters
consoleIO.txt	copy of the RKM console I/O chapter
diskfont.txt	warning of disk font loading bug
fullfunc.txt	list of #defines, macros, functions
inputdev.txt	preliminary copy of the input device chapter

License information on Workbench distribution license

printer pre-release copy of the chapter on printer drivers, from RKM 1.1 v11fd.txt 'diff' of .fd file changes from version 1.0 to 1.1

v28v1.diff 'diff' of include file changes from version 28 to 1.0

AMICUS Disk 5 Files from the Amiga Link / Amiga Information Network

Note that some of these files are old, and refer to older versions of the operating system. These files are from Amiga Link. For a time, Commodore supported Amiga Link, aka AIN, for online developer technical support. It was only up and running for several weeks. These files do not carry a warranty, and are for educational purposes only. Of course, that's not to say they don't work.

A demo of Intuition menus called 'menudemo', In C source

whereis.c	find a file searching all subdirectories
bobtest.c	BOB programming example
sweep.c	sound synthesis example

Assembler files:

mydev.asm	sample device driver
mylib.asm	sample library example
mylib.i	
mydev.i	
asmsupp.i	
macros.i	assembler include files

Texts:

amigatricks	tips on CLI commands
extdisk	external disk specification
gameport	game port spec
parallel	parallel port spec
serial	serial port spec
v1.1update	list of new features in version 1.1
v1.1h.txt	'diff' of include file changes frontoversion

Files for building your own printer drivers, including dospecial.c, epsondata.c, init.asm, printer.c, printer.link, printertag.asm, render.c, and wait.asm. This disk does contain a number of files describing the IFF specification. These are not the latest and greatest files, but remain here for historical purposes. They include text files and C source examples. The latest IFF spec is elsewhere in this library.

AMICUS Disk 6 IFF Pictures

This disk includes the DPSlide program, which can view a given series of IFF pictures, and the 'showpic' program, which can view each file at the click of an icon. The pictures include a screen from ArticFox, a Degas dancer, the guys at Electronic Arts, a gorilla, horses, King Tut, a lighthouse, a screen from Marble Madness, the Bugs Bunny Martian, a still from an old movie, the Dire Straits moving company, a screen from Pinball Contruction Set, a TV newcaster, the PaintCan, a world map, a Porsche, a shuttle mission patch, a tyrannosaurus rex, a planet view, a VISA card, and a ten-speed.

AMICUS Disk 7 DigiView HAM demo picture disk

This disk has pictures from the DigiView hold-and-modify video digitizer. It includes the ladies with pencils and lollypops, the young girl, the bulldozer, the horse and buggy, the Byte cover, the dictionary page, the robot and Robert. This includes a program to view each picture separately, and all together as separate, slidable screens. The 'seeilbm' program, to turn any screen into an IFF picture.

AMICUS Disk 8

C programs:

Browse	view text files on a disk, using menus S-E-D
Crunch	removes comments and white space from C files, S-E
IconExec	EXECUTE a series of commands from Workbench S-E
PDScreen Dump	dumps Rastport of highest screen to printer
SetAlternate	sets a second image for an icon, when clicked once S-E
SetWindow	makes windows for a CLI program to run under Workbench S-E
SmallClock	a small digital clockin a window menu bar
Scrimper	the screen printer in the fourth AC S-E

Amiga Basic Programs:

(Note: Many of these programs are present on AMICUS Disk 1. Several of these were converted to Amiga Basic, and are included here.)

AddressBook	a simple address book database
Ball	draws a ball
Cload	program to convert Compuserve hex files to binary, S-D
Clue	the game, Intuition driven
ColorArt	art drawing program
DeluxeDraw	the drawing program in the 3rd AC, S-D
Eliza	conversational computer psychologist
Othello	the game, as known as 'go'
RatMaze	3D ratmaze game
ROR	boggling graphics demo
Shuttle	draws 3D pictures of the space shuttle
Spelling	simple spelling program
YoYo	wierd zero-gravity yo-yo demo, tracks yo-yo to the mouse

Executable programs:

3Dcube	Modula-2 demo of a rotating cube
AltIcon	sets a second icon image, displayed when the icon is clicked
AmigaSpell	a slow but simple spell checker, E-D
arc	the ARC file compression progra must-have for telecom, E-D
Bertrand	graphics demo
disksalvage	prog. to rescue trashed disks, E-D
KwikCopy	a quick but nasty disk copy program: ignores errors, E-D
LibDir	lists hunks in an object file E-D
SaveILBM	saves any screen as IFF pic.E-D ??
ScreenDump	shareware screen dump prog, E only
StarTerm	version 2.0, term program, XmodemE-D

Texts:

LatticeMain	tips on fixing _main.c in Lattice
GDiskDrive	make your own 5 1/4 drive
GuruMed	explains the Guru numbers
Lat3.03bugs	bug list of Lattice C version 3.03
MForgeRev	user's view of the MicroForge HD
PrintSpooler	EXECUTE-based print spool prog.

.BMAP files:

These are the necessary links between Amiga Basic and the system libraries. To take advantage of the Amiga's capabilities in Basic, you need these files. BMAPs are included for 'clist', 'console', 'diskfont', 'exec', 'icon', 'intuition', 'layers', 'mathffp', mathieeedoubas', 'mathieeesingbas', 'mathtrans', 'potgo', 'timer' and 'translator'.

AMICUS Disk 9

Amiga Basic Programs:

FlightSim	simple flight simulator program
HuePalette	explains Hue, Saturation, & Intensity
Requester	ex. of requesters from Amiga Basic
ScrollDemo	demonstrates scrolling capabilities
Synthesizer	sound program
WorldMap	draws a map of the world

Executable programs:

Boing!	latest Boing! demo,with selectable speed,E
Brush2C	converts an IFF brush to C data instructions, initialization code, E
Brush2Icon	converts IFF brush to an icon, E
Dazzle	graphics demo, tracks to mouse, E
DeciGEL	assembler program for stopping 68010 errors, S-E-D
Klock	menu-bar clock and date display, E
life	the game of life, E
TimeSet	Intuition-based way to set the time & date
EMEmacs	another Emacs, more oriented to word processing, S-E-D
MyCLI	a CLI shell, works without the Workbench, S-E-D

Texts:

FnctnKeys	read function keys from Amiga Basic
HackerSln	explains how to win the game 'hacker'
Ist68010	guide to installing a 68010 in your Amiga
Boing!	latest Boing! demo,with selectablespeed, E
Brush2C	converts an IFF brush to C data instructions, initialization code, E
Brush2Icon	converts IFF brush to an icon, E
Dazzle	graphics demo, tracks to mouse, E
DeciGEL	assembler program for stopping 68010 errors, S-E-D
Klock	menu-bar clock and date display, E
life	the game of life, E
TimeSet	Intuition-based way to set the time date,
EMEmacs	another Emacs, more oriented to word processing, S-E-D
MyCLI	a CLI shell, works without the Workbench, S-E-D

Texts:

FnctnKeys	explains how to read function keys from Amiga Basic
HackerSln	explains how to win the game 'hacker'
Ist68010	guide to installing a 68010 in your Amiga
PrinterTip	sending escape sequences to your printer
StartupTip	tips on setting up your startup-sequence file
XfmrReview	list of Transformer programs that work

Printer Drivers:

Printer drivers for the Canon PJ-1080A, the C Itoh Prowriter, an improved Epson driver that eliminates streaking, the Epson LQ-800, the Gemini Star-10, the NEC 8025A, the Okidata ML-92, the Panasonic KX-P10xx family, and the Smith-Corona D300, with a document describing the installation process.

AMICUS Disk 10 Instrument sound demos

This is an icon-driven demo, circulated to many dealers. It includes the sounds of an acoustic guitar, an alarm, a banjo, a bass guitar, a boink, a calliope, a car horn, claves, water drip, electric guitar, a flute, a harp arpegio, a kickdrum, a marimba, a organ minor chord, people talking, pigs, a pipe organ, a Rhodes piano, a saxophone, a sitar, a snare drum, a steel drum, bells, a vibrophone, a violin, a wailing guitar, a horse whinny, and a whistle.

AMICUS Disk 11

C programs

dirutil	Intuition-based, CLI replacement manager S-E
cpri	shows and adjusts priority of CLI processes, S-E
ps	shows info on CLI processes, S-E
vidtex	displays Compuserve RLE pics, S-E

AmigaBasic programs

pointered	pointer and sprite editor program
optimize	optimization ex ample from AC article
calendar	large, animated calendar, diary and date book program
amortize	loan amortizations
brushtoBOB	converts small IFF brushes to AmigaBasic BOB OBJECTS
grids	draw and play waveforms
hilbert	draws Hilbert curves
madlib	mad lib story generator
mailtalk	talking mailing list program
meadows3D	3D graphics program, from A C™ article
mousetrack	mouse tracking example in hires mode
slot	slot machine game
tictactoe	the game
switch	pachinko-like game
weird	makes strange sounds

Executable programs

cp	unix-like copy command, E
cls	screen clear, S-E
diff	unix-like stream editor uses 'diff' output to fix files
pm	chart recorder performances indicator

Assembler programs

cls	screen clear and CLI arguments example

Modula-2

trails	moving-worm graphics demo
caseconvert	converts Modula-2 keywords to uppercase
Forth	Breshehan circle algorithm example
Analyze	12 templates for the spreadsheet Analyze

There are four programs here that read Commodore 64 picture files. They can translate Koala Pad, Doodle, Print Shop and News Room graphics to IFF format. Getting the files from your C-64 to your Amiga is the hard part.

AMICUS Disk 12

Executable programs

blink	'alink' compatible linker, but faster, E-D
clean	spins the disk for disk cleaners, E-D
epsonset	sends Epson setttings to PAR from menu E-D
showbig	view hi-res pics in low-res superbitmap, E-D
speaktime	tell the time, E-D
undelete	undeletes a file, E-D
cnvapidhm	converts Apple][low, medium and high res pictures to IFF, E-D
menued	menu editor produces C code for menus, E-D
quick	quick disk-to-disk nibble copier, E-D
quickEA	copies Electronic Arts disks, removes protection, E-D
txed 1.3	demo of text editor from Microsmiths,E-D

C programs

spin3	rotating blocks graphics demo, S-E-D
popcli	start a new CLI at the press of a button, like Sidekick, S-E-D
vsprite	VSprite example code from Commodore, S-E-D
AmigaBBS	Amiga Basic bulletin board prog., S-D

Assembler programs

star10	makes star fields like Star Trek intro,S-E-D

Pictures

Mount Mandelbrot	3D view of Mandelbrot set
Star Destroyer	hi-res Star Wars starship
Robot	robot arm grabbing a cylinder

Texts

vendors	Amiga vendors, names, addresses
cardco	fixes to early Cardco memory boards
cinclude	cross-reference to C include files
mindwalker	clues to playing the game well
slideshow	make your own slideshows from the Kaleidoscope disk

AMICUS Disk 13

Amiga Basic programs

Routines from Carolyn Scheppner of CBM Tech Support, to read and display IFF pictures from Amiga Basic. With documentation. Also included is a program to do screen prints in Amiga Basic, and the newest BMAP files, with a corrected ConvertFD program. With example pictures, and the SaveILBM screen capture program.

Routines to load and play FutureSound and IFF sound files from Amiga Basic, by John Foust for Applied Visions. With documentation and C and assembler source for writing your own libraries, and interfacing C to assembler in libraries. With example sound.

Executable programs

gravity	Sci Amer Jan 86 gravitation graphic simulation, S-E-D

Texts

MIDI	make your own MIDI instrument interface, with documentation and a hi-res schematic picture.

AMICUS Disk 14

Several programs from Amazing Computing issues:

Tools

Dan Kary's	C structure index program, S-E-D

Amiga Basic programs:

BMAP Reader	by Tim Jones
IFFBrush2BOB	by Mike Swinger
AutoRequester	example
DOSHelper	Windowed help system for CLI commands, S-E-D
PETrans	translates PET ASCII files to ASCII files, S-E-D
C Squared	Graphics program from Scientific American, Sept 86, S-E-D
crlf	adds or removes carriage returns from files, S-E-D
dpdecode	decrypts Deluxe Paint, remo
ves copy	protection, E-D
queryWB	asks Yes or No from the user returns exit code, S-E
vc	VisiCalc type spreadsheet, no mouse control, E-D
view	views text files with window and
slider	gadget, E-D

Oing, Sproing, yaBoing, Zoing are sprite-based
Boing! style demos, S-E-D

CLIClock, sClock, wClock are window border clocks, S-E-D

Texts

An article on long-persistance phospor monitors, tips on making brushes of odd shapes in Deluxe Paint, and recommendations on icon interfaces from Commodore-Amiga.

AMICUS Disk 15

The C programs include:

'pr'	a file printing utility, which can print files in the background, and with line numbers and control character filtering.
'fm'	displays a chart of the blocks allocated on a disk.
'Ask'	questions an 'execute' file, returns an error code to control the execution in that batch file
'Stat'	an enhanced version of AmigaDOS 'status' command.
'Dissolve'	random-dot dissolve demo displays IFF picture slowly, dot by dot, in a random fashion.
'PopCLI2'	invoke new CLI window at the press of a key.

The executable programs include:

'Form'	file formatting program through the printer driver to select print styles
'DiskCat'	catalogs disks, maintains, sorts,merges lists of disk files
'PSound'	SunRize Industries' sampled sound editor & recorder
'Iconmaker'	makes icons for most programs
'Fractals'	draws great fractal seascapes and mountain scapes.
'3D Breakout'	3D glasses, create breakout in a new dimension
'AmigaMonitor'	displays lists of open files,
memory use,	tasks, devices and ports in use.
'Cosmoroids'	version of 'asteroids' for the Amiga.
'Sizzlers'	high resolution graphics demo written in Modula 2.

Texts:

'ansi.txt'	explains escape sequences the CON: device responds to.
'FKey'	includes template for making paper to sit in the tray at the top of the Amiga keyboard.
'Spawn'	programmer's document from Commodore Amiga, describs ways to use the Amiga's multitasking capabilities in your own programs.

AmigaBasic programs:

'Grids'	draw sound waveforms, and hear them played.
'Light'	a version of the Tron light-cycle video game.
'MigaSol'	a game of solitaire.
'Stats'	program to calculate batting averages
'Money'	"try to grab all the bags of money that you can."

AMICUS 15 also includes two beautiful IFF pictures, of the enemy walkers from the ice planet in Star Wars, and a picture of a cheetah.

AMICUS Disk 16

'juggler' demo by Eric Graham. a robot juggler bouncing three mirrored balls, with sound effects. Twenty-four frames of HAM animation are flipped quickly to produce this image. You control the speed of the juggling. The author's documentation hints that this program might someday be available as a product.

IFF pictures

parodies of the covers of Amiga World and Amazing Computing magazines.

C programs:

'Inputhandler'	example of making an input handler.
'FileZap3'	binary file editing program
'ShowPrint'	displays IFF picture, and prints it.
'Gen'	program indexes and retrieves C structures and variables declared in the Amiga include file system.

Executable Programs:

'FixHunk2'	repairs an executable program file for expanded memory
'ms2smus'	converts Music Studio files to IFF standard 'SMUS' format. I have heard this program might have a few bugs, especially in regards to very long songs, but it works in most cases.
'Missile'	Amiga version of the 'Missile Command' video game,

This disk also contains several files of scenarios for Amiga Flight Simulator II. By putting one of these seven files on a blank disk, and inserting it in the drive after performing a special command in this game, a number of interesting locations are preset into the Flight Simulator program. For example, one scenario places your plane on Alcatraz, while another puts you in Central Park

AMICUS Disk 17

Telcommunications disk whuch contains six terminal programs.

"Comm" V1.33	term prog. with Xmodem, WXmodem,
"ATerm" V7.2	term prog. includes Super Kermit
"VT-100"V2.6	Dave Wecker's VT-100 emulator with Xmodem,Kermit, and scripting
"Amiga Kermit"	V4D(060) port of the Unix C-Kermit
"VTek" V2.3.1	Tektronix graphics terminal emulator based on the VT-100 prog. V2.3 and contains latest 'arc' file compression
"AmigaHost"	V0.9 for Compuserve. Includes RLE graphics abilities & CIS-B file transfer protocol.
"FixHunk"	expansion memory necessity
"FixObj"	removes garbage characters from modem received files
"Txt"	filters text files from other systems to be read by the Amiga E.C.
"addmem"	executeable version for use with mem expansion article in AC v2..1
'arc'	file documentation and a basic tutorial on un 'arc'ing files
"arcre"	for makeing "arc" files E.C.

AMICUS Disk 18

Logo	Amiga version of the populacomputer language, with example programs, E-D
Tv*Text	Demo version of the TV*Text character generator
PageSetter	Freely distributable versions of the updated PagePrint and PageIFF programs for the PageSetter desktop publishing package.
FullWindow	Resizes any CLI window using only CLI commands, E-D
Life3d	3-D version of Conway's LIFE program, E-D
Defdisk	CLI utility to re-assign a new Workbench disk, S-E-D
Calendar.WKS	Lotus-compatible worksheet that makes calendars
SetKey	Demo of keyboard key re-programmer, with IFF picture to make function key labels, E-D
VPG	Video pattern generator for aligning monitors, E-D
HP-10C	Hewlett-Packard-like calculator, E-D
SetPrefs	Change the Preferences settings on the fly, in C, S-E-D
StarProbe	Program studies stellar evolution. C source included for Amiga and MS-DOS, S-E-D
ROT	C version of Colin French's AmigaBasic ROT program from Amazing Computing. ROT edits and displays polygons to create three dimensional objects. Up to 24 frames of animation can be created and displayed. E-D
Scat	Like Ing, windows on screen run away from the mouse, E-D
DK	Decays" the CLI window into dust, in Modula 2, S-E-D
DropShadow2	Adds layered shadows to Workbench windows, E-D

AMICUS Disk 19

This disk carries several programs from Amazing Computing. The IFF pictures on this disk include the Amiga Wake part T-shirt logo, a sixteen-color hi-res image of Andy Griffith, and five Amiga Live! pictures from the Amazing Stories episode that featured the Amiga.

Solve	Linear equation solver in assembly language, S-E-D
Gadgets	Bryan Catley's AmigaBasic tutorial,
Household	Bryan Catley's AmigaBasic household inventory program, S-D
Waveform	Jim Shields' Waveform WoAmigaBasic, S-D
DiskLib	John Kennan's AmigaBasic disk librarian program, S-D
Subscripts	Ivan Smith's AmigaBasic subscript example, S-D
String, Boolean	C programs and executables for Harriet Maybeck Tolly's Intuition tutorials, S-E-D
Skinny C	Bob Riemersma's example for making small C programs, S-E-D
COMAL.h	Make C look like COMAL header file,
EmacsKey	Makes Emacs function key definitions by Greg Douglas, S-D
AMon 1.1	Snoop on system resource use, E-D
BTE	Bard's Tale character editor, E-D
Size	CLI program shows the size of a given set of files, E-D
WinSize	CLI window utility resizes current window, S-E-D

AMICUS Disk 20

Compactor, Decoder Steve Michel AmigaBasic tools, S-D

BobEd	BOB and sprite editor written in C,S-E-D
SpriteMasterII	Sprite editor and animator by Brad Kiefer, E-D
BlitLab	Blitter chip exploration C program by Tomas Rokicki, S-E-D
FPic	Image processing program by Bob Bush loads and saves IFF images, changes them with several techniques, E-D
Bankn	Complete home banking program, balance your checkbook! E-D

AMICUS Disk 21

Target	Makes each mouse click sound like a gunshot, S-E-D
Sand	Simple game of sand that follows the mouse pointer, E-D
PropGadget	Harriet Maybeck Tolly's proportional gadget example, S-E
EHB	Checks to see if you have extra-half-bright graphics, S-E-D
Piano	Simple piano sound program
CelScripts	Makes cel animation scripts for Aegis Animator, in AmigaBasic

This disk has electronic catalogs for AMICUS disks 1 to 20 and Fish disks 1 to 80. They are viewed with the DiskCat program, included here.

AMICUS Disk 22

Cycles	Light cycle game, E-D
Show_PrintII	Views and prints IFF pictures, including larger than screen
PrtDrvGen2.3	Latest version of a printer driver generator
Animations	VideoScape animations of planes and boing ball
Garden	Makes fractal gardenscapes
BasicSorts	Examples of binary search and insertion sort in AmigaBasic

AMICUS Disk 23

An AMICUS disk completely dedicated to music on the Amiga. This disk contains two music players, songs, instruments, and players to bring the thrill of playing "Big Sound" on your Amiga

Instruments	a collection of 25 instruments for playing and creating music. The collection ranges from Cannon to Marimba
List INSTR	program to list the instruments DMCS will not load as well as list the origins for any instrument.
Music	a collection of 14 Classical pieces
1812Overture	The 16 minute classical feature complete with Cannon!

Three Amiga Music Players:
SMUSPlay
MusicCraft2SMUS
MusicStudio2SMUS

AMICUS Disk 24

Sectorama	A disk sector editor for any AmigaDOS file-structured device, recover files from a trashed hard disk. By David Joiner of MicroIllusions
Iconize	Reduces the size of IFF images, companion program, Recolor, remaps the palette colors of one picture to use the palette colors of another. Using these programs and a tool to convert IFF brushes to Workbench icons, make icons look like miniatures of the pictures.
CodeDemo	Modula-2 program converts assembler object files to inline CODE statements. Comes with a screen scrolling example
AmiBug	Workbench hack makes the same fly walk across the screen at random intervals. Otherwise, completely harmless.
BNTools	Three examples of assembly language code from Bryce Nesbitt: 1. SetLace,prog to switch interlace on&off. 2. Why, replace AmigaDOS CLI Why 3. LoadIt, prog to load a file into memory until a reboot. (Only the most esoteric hackers will find LoadIt useful.)
Monolace	CLI program resets Preferences to several colors of monochrome & interlace screens. C source is included, works with DisplayPref, a CLI program which displays the current Preferences settings.
BoingMachine	A ray-traced animation of a perpetual motion Boing-making machine, includes the latest version of the Movie program, which has the ability to play sounds along with the animation. By Ken Offer
Daisy	Example of using the translator and narrator devices to make the Amiga talk. It is written in C.
QuickFlix	Script-driven animation and slideshow program flips through IFF images.
BMon	System monitor AmigaBasic program ; perform simple manipulations of memory.
Moose	Random background program, a small window opens with a moose resembleing Bullwinklesaying witty phrases user definable.
DGCS	Deluxe Grocery Construction Set, simple Intuition-based prog for assembling and printing a grocery list.

The Virus Check directory holds several programs relating to the software virus that came to the US from pirates in Europe as detailed in Amazing Computing V2.12. Bill Koester's full explanation of the virus code is included. One program checks for the software virus on a Workbench disk; the second program checks for the virus in memory, which could infect other disks.

AMICUS Disk 25

Nemesis	Graphics demo pans through space towards the mythical dark twin of the sun with wonderful music and space graphics.

The KickPlay directory holds text that describes several patches to the Kickstart disk. For Amiga 1000 hackers who feel comfortable patching a disk in hexadecimal, KickPlay offers the chance to automatically do an ADDMEM for old expansion memory, as well as the ability to change the picture of the "Insert Workbench" hand. A program is also included for restoring the correct checksum of the Kickstart disk.

KeyBird	BASIC prog edits keymaps, adjust the Workbench keymaps or create your own.

For PDS orders, please use form on page 112

SunMouse	This program automatically clicks in windows when the mouse is moved over them. V1.0, E-D

Fred Fish Disk 66

AmScsi	Preliminary plans for a SCSI disk controller board.
Asm68k	Macro assembler, version 1.0.1. E-D
Assigned	Example for avoiding DOS insert-disk requester, by scanning the list of 'assign'ed names. S-E-D
Dk	Pretends to eat away at CLI window. S-E-D
Flip	Flips whole screen as a joke. S-E-D
Foogol	Foogol cross-compiler generates VAX assembly code. S-E-D
Free	Prints amount of free space on all drives.S-E-D
MallocTest	malloc/free memory test program. S-E-D
Melt	Pretends to melt the screen. S-E-D
Nart	Graphic flying string demo. S-E-D
Purty	Easy way to set printer attributes from Workbench. E-D
RayTracer	Simple ray tracing program. E-D
SendPackets	Updated CBM examples of packet routines on disk 35. S-E-D
SnapShot	Memory resident screen dump. E-D
TagBBS	Shareware BBS system, version 1.02.

Fred Fish Disk 67

AmCat	Shareware disk cataloging program.
AmigaSpell	Shareware Intuition spelling checker, V2.0. E-D
Bouncer	3-D bouncing ball written in MultiForth, SED
Comm	Terminal program version 1.33, E
Dux5	Another version of DirUtil. S-E-D
HexCalc	Hex, octal, & decimal calculator. E-D
Icons	Various big and alternate image icons.
Mandala	Mandala graphics and sound. E
PersMait	Demo shareware personal file manager.
RSLClock	Menu bar clock version 1.3. E-D
RTCubes	Graphics demo of 3D cubes. E-D
Wheel	"Wheel of Fortune"-type game,AmigaBASIC

Fred Fish Disk 68

This is version MG 1b of the MicroGNUEmacs. Source and executable areincluded, as well as source for other computers besides the Amiga.

Fred Fish Disk 69

Asm68k	Macro assembler, v1.0.3, E-D
BlitLab	Blitter exploring program, in C, S-E-D
Conman	Replacement console device handler adds editing and history to any application that uses CON:, v0.9, E-D
Console	Replacement console routines, in C, S-E-D
Dk	Decays the screen bit by bit, update to disk 66, in Modula-2, S-E-D
Frags	Displays memory fragmentationby listing the size of free memory blocks, in C, S-E-D
IconType	Change the type of an icon, in C, S-E-D
Make	'make' in Manx C, S-E-D
MonProc	Monitors processes for packet activity, in C,S-ED
MouseClock	Mouse pointer into a digital clock,in C,SED
Sb	Browses system structures, from Transactor magazine, v1.0, in C, S-E-D
Spew	Generates 'National Enquirer'-type headlines from rules file. In C,S-E-D
Spool	Three programs to demonstrate multitasking & spooling in a printer spooler. In C, v1.2, S-E-D
Wc	Counts words ala Unix 'wc', but faster, in C,S-ED

Fred Fish Disk 70

This is a disk of shareware programs.

AmigaMonitor	Explores state of the system, v1.13
Arc	Standard file compressor andl librarian, v0.23, a port of MS-DOS v5.0. E-D
BlackBook	Phone book program.
DoTil	Intuition-driven file manipulator program,v2.0.
GravityWars	Game of planets, ships and black holes,v1.03.
Jobs	Alternate user interface to CLI and WB, v2.1.
Lens	Magnifies area around mouse, shows it in a window, v1.0.
Life-3d	3D version of the classic cellular-automaton game, v1.2.
Logo	Logo language interpreter
SetKey	Demo keymap editor, v1.0
Vpg	Makes displays for aligning video monitors, v1.0.

Fred Fish Disk 71

AirFoil	Makes airfoils using the Joukowski transformation, in C, S-E-D
Amiga Basic	Miscellaneous programs including 3D plot program, a kaleidoscope, C-A logo drawing program file comparison utility string search program, S-E-D
Blocks	A variation of "lines", but with variable color blocks. E-D
Comm	Great terminal program, v1.34, E-D
DiskX	Utility for exploring file system.E-D
Fpic	Simple image processing program that operates on IFF pictures, with several filters, merging images, E-D
IconMk	Makes icons for files , v1.2a, E-D
Icons	New icons
NewFonts	Two new fonts; 'shalt18', an electronic ciruit element font, and 'ibm5', a PC-like font.
PetCLI	An AmigaBASIC CLI shell program.
PWDemo	Demo of the commercial product PowerWindows,v1.2. It aids creation of custom windows, menus, and gadgets, giving C or assembly source. E-D
Rot	Creates and animates 3-D objects, v0.5, E,D
TimeSet	Sets time from Workbench, E-D

Fred Fish Disk 72

This is a disk of IFF pictures.

Fred Fish Disk 73

Add	Customizes existing program menus with Amiga-key shortcuts. Also includes 'until', which waits until a given window is created. Shareware, in C, S-E-D.
AutoIconOpen	Fools WB into thinking mouse has double-clicked icons. In C, S-E-D
Dio	Generic Exec device interface code for opening libraries, getting multiple I/O channels, asynchronous operations, etc. In C, S-E-D.
Dissolve	Slowly displays IFF files, ala Nov 86 Dr. Dobb's program. In C, S-E-D
DTerm	Flexible, reprogrammable terminal program v1.10, E-D
Expose	Re-arranges windows so that at least one pixel of menu bar gadgets are exposed. InC, S-E-D.
Lit	Scans a text file, converts to C-style printable strings.C,v2.0, S-E-D
Lmv	"Long Movie", program views series of IFF picts in quick succession, upto 19 fps. Shareware, E-D
MouseOff	Mouse pointer disappears after ten seconds of non-use. In C, S-E-D
ParOut	Examples of controlling parallel port with resources instead of the PAR: device.In C,S-E-D
PenPalFont	Child-like font.
RunBackGround	Similar to RunBack on disk 66, runs program from the CLI allowing the CLI window to close. In C,S-E-D
SnapShot	Screendump utility,update FF 66.E-D
TypeAndTell	Example installs a device handler before Intuition, and speaks each key as it is pressed. In C and assembler, S-E-D
Xplor	Prints info about system lists, in assembler,S-E-D

Fred Fish Disk 74

Cled	Edits and recalls CLI commands, v1.3, E-D
Control	Intercepts graphic printer dump calls and accesses color map, width,and screen resolution. C,S-E-D
Dme	Simple WYSIWYG text editor for programmers,v1.25. Update of FF 59.E-D
DropShadow	WB dropshadows, v2.0.UpdateFF59. E-D
Funds	AmigaBASIC prog tracks mutual or stocks-D
Less	Text viewing program, like Unix "more", v1.1, update to disk 34. S-E-D
Makemake	Scans C source files and constructs a vanilla 'makefile' in the current directory. S-E-D
mCAD	Object-oriented drawing prog. v1.2.4, update to FF 59.Shareware, E-D
Random	Simple random number generator in C. S-E-D
TDebug	Monitors devices by intercepting Exec SendIO() and DoIO() vectors, in C, v1.0, S-E-D
Units	Converts measurements in different units, includes "chart" option, in C, S-E-D
XCopy	Replacement for AmigaDOS 'copy', doesn't change the date, uses Unix wildcards. E-D

Fred Fish Disk 75

Bezier	Play with Bezier curves points and granularity, S-E-D
BSplines	Play with b-splines, as above, S-E-D
Comm	C source for Comm terminal program v1.34. S-E-D
Copy	Replacement 'copy' command v1.0, preserves date, in C, S-E-D
Diff	Simple 'diff' in C, S-E-D
DuM2	Another DirUtil in Modula-2, v1.5, S-E-D
Eless	Fast 'dir' program in C, S-E-D
Fd	Faster 'eless' in C, S-E-D
HardCopy	Sends a transcript of a CLI session to a file, in C, S-E-D
MouseOff	Update FF73, turns off mouse pointer, S-E-D
SetFont	Changes the font in a Workbench screen, v2.0, S-E-D
SpeedDir	Another fast 'dir', in assembler, S-E-D

Fred Fish Disk 76 & 77

These are disks 1 and 2 of Chris Gray's Draco distribution for the Amiga. Draco is a compiled, structured language reminiscent of both C and Pascal. A full interface to AmigaDOS and Intuition is supplied. Be sure to get both disk 76 and 77.

Fred Fish Disk 78

Cycles	Cycle game like "Tron", v1.0, E-D
EOMS	Experts Only Mercenary Simulator game, E-D
MandelVroom	Mandelbrot generator with enhanced palette controls, fixed/floating point, presets, v1.50, in Manx C, S-E-D

Fred Fish Disk 79

AsmTools	CLI tools in assembler: echo, loadit, mounted, setlace, why; S-E-D
AssignDev	Give devices multiple names, in C, S-E-D
AuxHandler	Example of a dos handler that allows use of a CLI via the serial port. Includes source. Author: Steve Drew
Cmd	Redirects printer output to a file, in C, S-E-D
Info	AmigaDOS 'info' replacement, in C and assembler, S-E-D
Kill	Removes a task and its resources, in C,S-E-D
M2Error	Displays errors from TDI Modula-2 compiles, S-E-D
MonProc	Update to process packet prog. from FF69,in C,S,E,D
Mounted	Program for testing if a drive is present, in a script In C, S-E-D
Nro	Another 'roff'-style text formatter, in C, S-E-D
ParTask	Finds parent task, in C, S-E-D
QueryAny	For scripts, asks a question, accepts Y/N, gives return code. In assembler, S-E-D
ScnSizer	Resets pref settings for screen size, in C,SED
SharedLib	Example, shared lib, in C & assembler, S-E-D
Task	Simple CreateTask() example in C, S-E-D
Uw	Unix Windows client v1.0, in C, S-E-D
Who	Lists tasks on ready and wait queues, in C, S-E-D

Fred Fish Disk 80 (see Fred Fish 90)

Fred Fish 80 has been withdrawn due to copyright problems.

Fred Fish Disk 81

Asm68k	V1.1.0 of a macro assembler
AutoFacc	Shrinks the FACC window and moves it to the back
Brushes	53 custom IFF brushes of electronic symbols
CheckIFF	Checks structure of an IFF file CledV1.4 update FF74 of a simple CLI
Conman	Replaces console handler to add editing and history to many programs
Fonts	Miscellaneous fonts
Icon	V6.0 of the Icon programming language
KeyLock	Freezes the keyboard and mouse until pass word entered.
ScatDisplay	hack created from "lng"
Smush	Smushes an IFF file.
Target	Each mouse clikc becomes a gunshot

Fred Fish Disk 82

Adventure	Port of the classic Crowther and Woods game
AmicTerm	V0.50 of a telecommunications program, with scripts, redial, beeps, enhanced file requester
D2D-Demo	Demo version of Disk-2-Disk by Central Coast Software
DX-Synth	Voice filer program for Yamaha DX series synthesizers, update to disk 38
DiskMan	V1.0 of another DirUtil program
Icons	Miscellaneous new icons
Panl	Universal MIDI patch panel, v1.2
Rocket	Another Workbench hack, plays Lunar Lander
Sand	Game of sands following your pointer.

Fred Fish Disk 83

This disk contains a demo version of TeX from N Squared. It is limited to small files, and the previewer can only display ten pages or less, and only a small number of fonts are provided.

Fred Fish Disk 84

AudioToolsPrograms from Rob Peck's July/August Amiga World article	
BlitLab	Blitter experimentation program, V1.2, update to FF69
Ed	Simple editor, similar to Unix 'ed', based on the editor in Software Tools.
GravityWars	Game of planets, ships and black holes, v1.04, update to disk 70.
HunkPad	Adds legal padding to executables for Xmodem transmission.
PipeHandler	An AmigaDOS pipe device which supports named pipes and taps. V1.2
PopCLI	V3.0 of a hot-key to invoke a CLI window, with screen blanker, update to disk 40.
Requester	Update FF34, file requester similar to DPaint.
ScottDevice	V33.1 of a 'mount'able MicroForge SCSI driver.
Viacom	Another Schwab hack, makes TV-like static on screen.Parody

Fred Fish Disk 85

Csh	V2.06 of Dillon's 'csh'-like shell
FileReq	Source to wildcard file requester
Hide	Hides expansion memory from programs
ImageTools	Shareware tools to manipulation IFF images
LowMem	ServerShared library to aid in low memory situations
Plot6	A star plotting program with source.
RawIO	Example of setting raw mode on standard input
Rocket	Lunary Lander for Workbench, with source.
VMore	"more"-like text viewing utility, v1.0 SE
Vnews	Simple Unix news reader.

Fred Fish Disk 86

AutoPointAuto-selects window under the mouse pointer, with screensaver.	
ClickToFront	Double-clicks in window brings it to front, v1.1, S-E-D
Cmd	V3.0 of a tool to redirect printer output toa file.
FileIISG-Demo	Demo of Softwood File IIsg, a database manager with sound and graphics.

Fred Fish Disk 87

AdvSys	Adventure system from Byte May 1987, v1.2 E-D
AutoIconOpen	Fools Workbench to open disk icons, V1.2 update to disk 73, S-E-D
Claz	Converts IFF files to PostScript, V2.0, SED
Commodi	tiesMackraz's Commodities Exchange, an exec library to manage input handler, v0.4
Diff	Update to disk 75 of Unix-like 'diff', S-E-D
Dme	V1.27 of Dillon's text editor, update FF74,E-D
DropShadow	V2.0 of prog. that puts shadows on Workbench, S-E-D
Elib	Shared library example in Manx C.
ID-Handler	An AmigaDOS device handler generates unique identifiers, V1.0, S-E-D
Install	Alternate AmigaDOS 'install' programs,SED
MemWatch	Waits for low memory trashing, V2.0, SED
MovePointer	Moves pointer to given location, S-E-D
MoveWindow	Move window to given location, S-E-D
MunchingSq	Munching Squares hack, S-E-D
PalTest	Test to see if this is a PAL machine, S-E-D
Sc	Generates random scenery, S-E-D
Tek4695	Tek4695 printer driver
WBDualPF	Example of dual-playfield screen, update FF41, S-E-D
WarpText	Fast text rendering routines, S-E-D
Yaiff	rExample IFF reader, S-E-D
Zoo	A file archiver like "arc", v1.42A, E-D

Fred Fish Disk 88 (see Fred Fish 89)

FF Disk 88 has been removed due to copyright problems

Fred Fish Disk 89 (replaces Fred Fish 80)

DirMaster	Disk catalogue program, V1.0a, E-D
FuncKey	Shareware function key editor, V1.01, E-D
MFF-Demo	Demo of MicroFiche Filer database prog
ScreenShift	Adjust screen position ie Preferences,SED
Snake	Bouncing squiggly lines demo, S-E-D
AutoEnguirer	screen contraption requester improvement S-E-D
DemoLition	Display Hack S-E-D

Fred Fish Disk 90 (replaces Fred Fish 80)

AmiGazer	Night sky viewer of 1573 stars, set date, time, day. E-D
CardFile	AmigaBasic card file study aid. E-D
Conman	Console handler replacement gives line editing and history to most progs, v0.98,ED
IMandelVroom	Slight update to disk 78 Mandelbrot program, E-D
NewDemos	Replacements for lines and boxes demos that take less CPU time, E-D
Othello	Game of Othello, E-D
PrinText	Displays text files with gadgets, speech, IFF display, v1.2, E-D
PrtDrvGen	Automatic printer driv. generator,v2.2b,ED
RainBench	Cycles colors of WB backdrop or text. ED
ShortCut	Makes single-key shortcuts for entering commonly typed CLI commands,&custom macros. E-D
ShowPrint	Displays and prints all sizes of IFF pictures & controls printer output styles, v2.0 E-D
Sizzlers	Graphics demos, v1.7.0, E-D
Timer	Small Workbench timer counts time and $/ minute, E-D

Fred Fish Disk 91

Adventure Definition Language (ADL) a superset of an older language called DDL by Michael Urban, Chris Kostanick, Michael Stein, Bruce Adler, and Warren Usui. ADL enhancements by Ross Cunniff. Included are sources to the ADL compiler, interpreter, and debugger. Binaries combined by Ross with Lattice 3.03. CLI enviornment only. Documentation is available from the authors.

Fred Fish Disk 92

As6502	portable 6502 assembler, C source, by J. Van Ornum, Amiga port by Joel Swank
Bawk	Text processor update from FF65 Inspired by UNIX awk. Searches files for patterns, performs actions based on patterns. By Bob Brodt; Amiga port by Johan Widen
HunkPad	update of FF84 version. by J. Hamilton,pads an object file to a multiple of 128 bytes for better xmodem transfer. S E
Less	Like Unix "more", better, version 1.2 update of FF74. Scrolls Back and f orward. S E by Mark Nudelman, Amiga port by Bob Leivian.
Ndir	Library that implements the 4BSD unix dir access routines by MikeMeyer. S
Parse	Recursive descent expression parcer, computes, and prints expressions. Includes transcendental function support. c Source included. by J Olsen
Shar	Two programs to pack and unpack shell archives includes C source. by Fabbian G. Dufoe
SmallLib	8 times smaller Amiga.lib replacement, binary only. by Bryce Nesbitt
UUencode	Encode/decode binary files for e mail or text-only methods. Update of FF53, includes checksum technique, compatible with older versions, plus transparent to older versions options. By Mark Horton, modified by Alan Rosenthal & Bryce Nesbitt.

Fred Fish Disk 93

Dme	Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, fast scro-lling, title-line statistics, multiple windows, ability to iconify windows. Update of FF87, SE. by Matt Dillon
MicroEmacs	Version 3.8, update to FF61 includes source. Orig by Dave Conroy modifications by Daniel Lawrence

Fred Fish Disk 94

AudioTools	Demo programs from Rob Peck's July/Augustt issue of AmigaWorld on accessing the audio device. V2,update of FF84. S. by Rob Peck
ClickUpFront	Similar in function to ClickToFront prog (FF86), bring windows tofront by clicking on any part of them. V 1.0. by Davide Cervone SE
HeliosMouse	Automatically activate a window simply by moving the mouse pointer into the window. V 1.0. Includes source. By Davide Cervone
IFF2Ps	Convert any IFF file to postscript for printing or viewing on a postscript compatible device. Version 1.2, by William Mason and Sam Paolucci E
ModulaTools	Various Modula 2 prog. routines. by Jerry Mack
Terrain3d	Pseudo-random 3d relief scenery generator, update of "sc", FF87. by Chris Gray, 3d by Howard Hull

Fred Fish Disk 95

Cmd	redirects the serial.device or parallel. device output to a file. Capture print jobs, debug or "offline" printing.V4 By C Scheppner SE
CygnusEdDemo	Demo of CygnusSoft's CygnusEd editor, a multiplefile, multiple feature editor.Includes demo 3.0 of MandFXP. by CygnusSoft Software E
Gomf	"Get Outa My Face" makes the Guru go away to allow clean-up & shutdown more cleanly. V 1.0, by Christian Johnsen E
Journal	records sequence of mouse & keyboard events, stored in a file for future playback. Good for demos or documententing bugs. E. by D. Cervone
MergeMem	attempts merging of MemList entries of sequentially configured ram boards. Allows allocating a section of memory which spans both boards. V 2, update of FF56. by Carolyn Scheppner SE
PrinterStealer	Asimilar to "Cmd", allows diversion of output destined for printer to a file. Binary only, Source avai. from authors. by A. Livshits & J-M Forgeas
Record-Replay	similar to "Journal", records and plays back mouse and keyboard events. B only. source avail. from authors, Alex Livshits & J-M Forgeas

Fred Fish Disk 96

AnimPlayer	Animation reader and displayer by the combined efforts of Videoscape, Sculpt3D, Silver, Forms- In-Flight, and Animator:Appreticeby M Hashetal.
Chess	Amiga port, non-Amiga interface. High playability. V 1.0, S. by J. Stanback, Amiga port by B. Leivian
Hackbench	provides source for WB-Likeprog, for experimentation & validation of new interface ideas. Not a WB replacement. by Bill Kinnersley
Label	Print labels with arbitray text. V 1.3, Source available from author, M.Hansen
LineDrawer	Produces line drawings based on drawing commands stored in a text file. Includes demo that draws an outline map of the USA and state borders. V1.0, SE. by John Olsen
PopUpMenu	Example code implementing pop-up menus, reasonably compatible with Intuition menus.SE. by Derek Zahn
Tek4695	Tektronix 4695/4696 printer driver. SE. by P Staub
TimeRam	Fast and Chip ram test prog,E by B Takahashi
WarpText	Fast text rendering routines, to be linked with application progs.Text display "as fast or faster than 'blitz'". V2.0 update of FF87, S by Bill Kelly

Fred Fish Disk 97

Replaces FF57 for Copywrite problems

CutAndPaste	Implementaions of Unix cut and paste commands. by John Weald
GraphIt	Progran to plot simple functions in 2 or 3 dimensions. by Flynn Fishman
Juggler	V1.2 of robot juggler animation. Uses HAM mode and ray tracing. by Eric Graham
MouseReader	Shareware program to read text files & view IFF files using only the mouse. by William Betz
Splines	Prog to demonstrate curve fitting & ren-dering techniques. by Helene (Lee) Taran
Shm	Graphics demo, approximately simulates the motion of two interacting pendulums. Includes S by Chris Ediss

Fred Fish Disk 98

Access	16 color terminal program based on Comm V1.,34. Includes Macro window, custom gadgets, colorized menus, etc. V. Beta 0.18 by Keith Young ,comm by D.J.James. E.
Backup	Writes AmigaDos disks as the backup destin-ation. recover files from the backup disk. Requires manual decisions on disk structure. by Alan Kent SE
DCDemo	DishCat 2.3, a disk catalog program, demo limited to cataloging 100 files at a time. by Ed Alford, MicroAce Software
HdDriver	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED
QBase	Quick-Base, a "MailBase Management utility", define and maintain a maximum of 200 records per file. by Kevin Harrise E
Thai	Thai language quiz program. Speak or type english/Thai sentences from supplied file. by Alan Kent SE

Fred Fish Disk 99

A-Render	Version .3 a Ray- Tracing Construction Set for the Amiga Computer by Brian Reed ED

Fred Fish Disk 100

Berserk	Must see animation, by Leo Schwab
Conman	Console handler replacement, provides line editing and command line histories transparent to application prog uses CON: windows. Shareware V1.0 by W Hawes. E.
WBLander	Workbench display hack game, upgrade of "Rocket" on FF85, now with sound effects. By Peter da Silva. E

Fred Fish Disk 101

CirPlane	Circular plane generator for VideoScape3D. Generates a clockwise circular polygon with the specified number of vertibles. V1.0 by T Floryan SE
IconAssembler	Change Workbench Icons with IFF-brush files by Stefan Lindahl E
Microspell	Standalone spelling checker scans text files and reports errors. 1000 common word list, 43,000 word main dictionary with multiple user dictionary support. Interfaces with MicroEMACS 3.9 with an emacs macro to step through the source file, stopping at suspect words and allowing the user to option. V1.0 by Daniel Lawrence, SED
Midi	midi library and utility set. Includes Midi monitor, routing utility, status utility, and more. by Bill Barton SED
PsIntrp	Postscript Interpreter reads and previews files on screen. by Greg Lee S(assy)E
StartUps	Three C startup file replacements for standard Astartup.obj and LStarup.obj. Options include (1) BothStartup.obj, for the WorkBench programs or CLI programs with or without command line parameters. (2) WBStartup.obj, for WorkBench programs or CLI programs that require no command line parameters. (3) CLIStartup.obj for CLI programs that require command line parameters but do not need to be WorkBench runnable. by Bryce Nesbitt SE

Fred Fish Disk 102

Dbug	Machine independent macro based C de-bugging package. Update FF41. by F Fish profiling support by Binayak Banerjee SE
Match-stuff	Heavy duty text pattern matching stuff. includes simple match text replacement capability. By Pete Goodeve
Sectorama	Recover lost or damaged data from floppy or hard disks or repair a damaged volume. by David Joiner E
SiliCon	Smart input line interpreter with window for full editing. Upgrade FF50 by P Goodeve., E
Xicon	Use icons to call up scripts containing CLI commands. V2.0 upgrade of FF31.by Pete Goodeve E

Fred Fish Disk 103

AvlTrees	Library and test prog. implement routines for creating and using trees held in memory.S.
Calc	A programmable RPN calculator.
Cref	A C cross ref. prog. S.
DosKwik	A pair of progs. allows you to save files to one or more floppies for quick loading. Doesn't store Dos format.
IntuiDos	A prog. to improve control and handling of the material on all disks in 'CLI-area' .
MFF-Update	A text import util. for MicroFiche Filer (demo on FF 89) and updates to some PD disk library datatbases.
Pack-It	Takes all files the files and dirs. on a disk & packs them into a single file. for modem.
Sol	Amiga version of solitaire.

Fred Fish Disk 104

Analyticalc	Is a large and powerful spreadsheet prog.

Fred Fish Disk 105

AsmProgs	Misc. assembly tools. Includes some S.
BasicProgs	LeastSquare solves least square probs ,graphs results,S.
Bison	A replacement for unix "yacc" command. S.
Dmouse	Another prog in the tradition of display hacks". S.
FlamKey	Allows keyboard and mouse inputs to be locked until a password is entered.
GravityWars	Game of planets,ships &black holes, v2.0 ,FF84 update.
IPo2C	A util. to write a C-lang definition to mimic the intuition pointer.S
Pere-et-Fil	Ex. of creating & using reentrant processes. S.
Record Replay	Similiar to "Journal" v2.0 update to FF95.

Fred Fish Disk 106

Funckey	Shareware function key editor, v1.1update to FF89. Source avail. from author(Anson Mah).
MoreArt	A small selection of some Amiga artwork.
QuickFlix	An IFF slideshow and cel animation prog,v0.13.
RistiNolla	A Finnish game. Also called Go-Moku. v1.0

Fred Fish Disk 107

Csh	V2.07 of Matt Dillon's csh like shell.S.
Diff	A util.,similiar to other common "diff" programs.S.
ProSuite	Provides ex. code of facilities such as FileIO Requester, XText, DoRequest. & tutorial on how to program the Amiga. Book 1.01.S
SVTools	Some useful tools. S.

Fred Fish Disk 108

AList	Dir listing prog. based on LD4 prg S
DirMaster	Disk cataloger. v1.0b, update to FF89. S.
Dots-Perfect	Printer Driver for an Epson MX80 printer with upgrade kit installed. S.
MonIDCMP	Lets you monitor the IntuiMessages that pass through an IDCMP window. Prints the message class,mouse coordinates,qualifier values. Great for debugging. S.
PrintPop	A util. to send common control settings to PRT: S.
Sectorama	Utilities to recover lost or damaged data from floppies & hard disks. v1.1, an update toFF102.
Tek	Vt100 emulator for a Tektronix 4010/4014. (V2.6) update to FF52. S.
Zoo	File archiver, like "arc". v1.24B. update to FF87

Fred Fish Disk 109

Machine	A new animation.
SimCPM	A CP/M sim,simulates 8080 along with h19 emulationS.
UUpc	Hook up your Amiga as a usenet node. S.

Fred Fish Disk 110

A68k	A 68000 assembler written in C. S.
Pdc	An optimizing C compiler for the 68000 processor. update to FF53. but not based on that code.

Fred Fish Disk 111

AmyLoad	A graphical monitor of cpu, blitter, & memory use. Includes two components; load.device,monitors system parameters, & amyload, which is the user interface & display program. by Jeff Kelley SE
AssignDev	Assigns multiple names to a given device. modified version of the original released on disk number 79. By Phillip Lindsay, mod by Olaf Seibert SE
Gauge	Continuously displays memory usage in a vertical bar graph Binary only. By:Peter da Silva
HeliosMouse	Another "sunmouse"prog. Automatically activates a window by mouse pointer V 1.1, update to FF94. By:Davide Cervone SE
Labels	Alphabetic & numeric ordered cross reference lists of defined system constants. Recommended for debugging purposes only, use the symbolic values in progs! By:Olaf Seibert
Mandel	Mandelbrot generator program, with bits & pieces of code from C. Heath & R.J. Mical. By:Olaf Seibert S
PopLife	A PopCLI type that plays life all over your screen. Lots of bits & pieces from Tomas Rokicki's blitlab & John Toebes' PopCLI. By:Olaf Seibert S

Fred Fish Disk 112

BeachBirds	Beach scene portrayed by sprites & sound 512K machine. By:Jerrold Tunnell B only.
Bully	Pushes all open screens around (thus the name "bully"). Show more than one demo at a time By:Mike Meyer S
DropShadow	Dropshadow V2.0, use with Bryce Nesbitt's Wavebench demo. B only. By:Jim Mackraz
HagenDemos	"RGB" & "Focus". RGB requires one meg, B only. By:Joel Hagen
Viacom	Latest version of viacom for use in conjuction with WaveBench demo. B only. By:Leo Schwab & Bryce Nesbitt
WaveBench	A neat screen hack, & runs on 512K machines. For more laughs, try in conjunction with Viacom or Ds (Dropshadow). Includes S. By:Bryce Nesbitt

Fred Fish Disk 113

AmiCron	Simple Unix "cron" type program,a background task uses a disk-resident table to automatically run certain tasks on a regular basis, at specific times. V 2.3, S. By:Steve Sampson, Amiga port by Rick Schaeffer
Dme	V 1.28f of Matt's text editor. A simple WYSIWYG editor for programmers. Not a WYSIWYG word processor Features : arbitrary key mapping, fast scrolling, title-line statistics multiple windows, iconify windows,etc. Update to FF93, S. By:Matt Dillon
DosDev	Example DOS device driver in Manx C. Version 1.10, includes S. By:Matt Dillon
M2Amiga	Demo of M2Amiga. A fast single pass Modula-2 compiler with editor, linker, a small set of interface & standard libraries. Compiles only small demo programs by limiting codesize & imports. Further development of the ETHZ compiler on FF24. B only. Demos with Source. By:R. Degen, C. Nieder, M. Schaub, J. Straube (AMSoft)
NoIconPos	Clears position info of any icons, allows WorkBench to pick a new place for the icon. Useful for disk & drawer icons where Snapshot rewrites the icon & the window information. Modula-2, another demo for M2Amiga By:Markus. Schaub

Fred Fish Disk 114

CDecl	English to C (and vice versa) translator for C declarations, a must for anyone except possibly the most hardcore C guru. By:Graham Ross, S
Vt100	V2.7 of vt100 terminal emulator with kermit & xmodem file transfer. Includes a few bug fixes posted to Usenet shortly after the posting of v2.7. Update to FF55. Includes S. By:Dave Wecker
WBLander	a special version of the WBLander program from FF100. Ending is unique. Effective use of sound, Includes S. By:Peter da Silva & Karl Lehenbauer

Fred Fish Disk 115

Killer	Masterful Video commercial of the Amiga, Beatles music, requires one meg of memory to run. Binary only. By:R. Wilt
Marketroid	Another devious sprite oriented demo with lots of "in" jokes. 512K required, includes S. By:Leo Schwab

Fred Fish Disk 116

Movies	A ram animation system with three different example animations; Kahnankas, Rocker, & F-15. Kahnankas & Rocker run on a 512K Amiga & show off overscan HAM mode. Includes a animation player program (movie), animation builder programs (dilbm, pilbm), & a text/graphics display program (vilbm). By:Eric Graham & Ken Offer

Fred Fish Disk 117

AMUC_Demo	A really neat horizontal scrolling demo that is a 2400 x 200 pixel 32 color IFF picture composed of digiview snapshots of members of the AMiga Users of Calgary, superimposed on a very wide picture of the Calgary Skyline. B only. By:Stephen Vermeulen & Stephen Jeans
ExP_Demo	Demo version of Express Paint 1.1., used to create the scrolling demo picture in the AMUC_Demo drawer on thi disk. B only. By:Stephen Vermeulen

Fred Fish Disk 118

Empire	Complete rewrite, in Draco, of Peter Langston's Empire . A multiplayer game of exploration, economics, war, etc, can last months.Use local keyboard or modem.V1.0, shareware, & S . By:Chris Gray, original game by Peter Langston
HAMmmm	Displays lines whose end points are bouncing around the screen, which is a double buffered HAM screen. The Y positions of the points are continuously copied into an audio waveform and played on all four channels, & the pitch of a just intoned chord is derived from the average X position of these points. JForth. Source By:Phil Burk
Stars	Based on original code by Leo Schwab, credits longer than actual demo. Runs on 512K Amiga. B only. By:Hobie Orris
WireDemo	Demonstrates the Amiga's line drawing speed. Runs on a 512K Amiga. Includes S. By:Matt Dillon

Fred Fish Disk 119

MicroEMACS	V3.9e of Daniel Lawrence's variant of Dave Conroy's microemacs. Update to FF93. Also included, for the first time, is extensive documentation in machine readable form. SE. Author: Dave Conroy, Enhanced by Daniel Lawrence

Fred Fish Disk 120

Amoeba	Clone of Space Invaders, one of the PDS games for the Amiga. B only BY: LateNight Developments
BackGammon	Graphical Backgammon (an undergraduate A.I. course project). Version 1.0, S. By: Robert Pfister
Bankn	A complete checkbook system offered by the author as shareware. Version 1.3, binary only. By: Hal Carter
EgyptianRun	"road race + hazards" type game. Version 1.1, B only, shareware, source available from author. By: Chris Hames
IconImage	Replace an old icon image with a new image, without affecting icontype, drawer data, etc. SE. By: Denis Green

Fred Fish Disk 121

BasicStrip	AmigaBASIC prog. helps to convert programs written in other forms of Basic to AmigaBASIC. By: George Trepal
DataPlot	Shareware,AmigaBASIC, plotting program. Also includes a least squares curve fit program. By: Dale Holt
Plot	Shareware 3-D AmigaBASIC graphing prog. & sample output plots. Source available via author.By: George Trepal
Stairs	AmigaBASIC prog. demos a musical illusion based upon perceptual circularity of widely spaced tones whose volumes are defined as a sinusoidal relationship to their frequency. By: Gary Cuba
Uedit	V2.3 of this nice shareware editor. With learn mode, command language, menu customization, and other user configurability and customizability features. Binary only, shareware, update toFF60. By: Rick Stiles
WBColors	Prog to change Workbench colors for progs that expect to be booted off their distribution disk but are run from a hard disk. SE. Author: Stefan Lindahl

Fred Fish Disk 122

Asteriods	Asteroid game. The images and sounds are replaceable by the end user. Anything goes! By: Rico Mariani
Iff2Pcs	Iinteractive puzzle prog. takes any IFF file with up to 16 colors, and breaks it up into squares to make a puzzle which the user can then piece together. V1.0,S. By: Ali Ozer
Names	A shareware program to create and manage mailing lists. Binary only. By: Ernie Nelson
Pr	Utility to print listings in different formats. Similar to the Unix "pr" program. Includes source By: Samuel Paolucci
PushOver	Board strategy game, AmigaBASIC. Push your pieces onto the board until yfive in a row in any direction.S. By: R.Yost
PuzzlePro	Create a puzzle from an IFF picture, which the user can then piece back together again. AmigaBASIC. V1.0, B only, shareware, source available from author. By: Syd Bolton

Fred Fish Disk 123

Arp	ARP stands for "AmigaDOS Replacement Project". Arp is an effort led by Charlie Heath of Microsmiths Inc., to replace the current DOS in a compatible fashion, so that current programs will continue to work. Arp also makes whatever improvements are possible, so that current and future programs will work better. Various authors contributed work
Car	One of Allen's entries to the Badge Killer Demo Contest. It apparently is an inside joke relating to a well known Amigan's experience with a certain highend graphics hardware manufacturer. Author: Allen Hastings

Fred Fish Disk 124

Icons	Some sample animated icons. By: L. Pfost
Tarot	AmigaBASIC Nice graphic of tarot cards. Author: L.Pfost

Fred Fish Disk 125

ElGato	Animation entry to the BK D Contest. Background music arrangement, requires Sonix to use. By Kevin Sullivans

Fred Fish Disk 126

Colour	Manipulate the colors of specific named screens, saving current color sets to data files, loading new color sets from data files, or interactively changing colors. S. By J. Russell
Dance	Two programs, "dancing polygons", are entries to the BKD Contest. They aresimilar, but demonstrate the range of colors available on the Amiga. S. By: John Olsen
HBHill	Animation entry to the BDK Contest. First known animation using the "Extra Half Brite" mode. By: Kevin Sullivan
Iconify	Subroutine creates an icon on the Amiga screen that can be subsequently dragged around, and double-clicked on. You can use this to have your programs "iconify" themselves to temporarily get out of the user's way. With source & demo program. By Leo Schwab
OnlyAmiga	Animation entry to BDK Contest. Three balls being juggled by pyramids rotating on their tops. By Iqbal Singh Hans
Suplib	Support library needed to rebuild various programs of Matt's from source, including DME, DTERM, etc. S By Matt Dillon
VCheck	V1.2 of virus detection prog. from Commodore Amiga Technical Support.. Will test for the presence of a virus in memory, or on specific disks. B only. By: Bill Koester.

Fred Fish Disk 127

Bounce	Entry for BDK Contest. Creates little dots that bounce around and multiply. S By: Steve Hansel and Tom Hansel
Nemesis	Entry to BDK Contest. It is quite small for what it does, and won fifth place in the contest. B only. By Mark Riley
Ripples	Entries to BDK Contest. Unlike most other animations, it shows a fixed object from a moving point of view, instead of vice versa. By: Allen Hastings

Fred Fish Disk 128

Dis	68000 disassembler, written in assembler. S By Greg Lee
DropCloth	Place a pattern, a 2 bitplane IFF image or a combination of a pattern and image, into the WorkBench backdrop. Version 2.2, shareware, B By: Eric Lavitsky
LedClock	An extremely simple clock program, for interlaced screens only. S By: Ali Ozer
MRBackUp	Hard disk backup utility, does a file by file copy on AmigaDOS floppy disks. With an intuition interface & file compression. V1.3, Source. By: Mark Rinfret
Paint	Simple screen painting program, written in web. Requires web preprocessing program to rebuild from source. Includes source in web. Author: Greg Lee
PrtDriver	A printer driver for the Toshiba "3 in one" printer in its Qume (best) mode. Includes source in C and assembler. By: Rico Mariani
SDBackUp	A hard disk backup utility. CLI interface only. Does file compression. V 1.1, binary only. By Steve Drew
Sed	A clone of the Unix sed (Stream EDitor) program. Includes source. By: Eric Raymondw
Keys	A "hot-keys" program binds keyboard function keys to window manipulation functions (window activation, front to back, moving screens, etc). S By: Davide Cervone

Fred Fish Disk 129

DosKwik	A pair of programs which allow you to save files, or a group of files, to one or more floppies for quick loading, does not store files in DOS format, for speed. V2.0, update to FF103. B, Shareware. By: Gary Kemper
MRBackUp	A hard disk backup utility, does a file by file copy to standard AmigaDOS floppy disks. Includes intuition interface & file compression. V2.0 (with sources) and 2.1 (binary only, source available from author). Update of FF128. By Mark Rinfret
PaintJet	HP PaintJet printer driver from HP sources.
Patch	Two independent ports of Unix utility "patch", which applies context diffs to text file to automatically update them. Patch V 1.3 was ported to the Amiga by Rick Coupland and patchV2.0 was ported by Johan Widen. S By: Larry Wall

Fred Fish Disk 130

DirMaster	Shareware disk cataloger, V1.1, update of FF108, new features and enhancements. B only. By Greg Peters
Evo	Human evolution toy/tutorial with source.By S. Bonner
Hp	RPN calculator prog, supports calculations with binary, octal, decimal, hex, float, and complex numbers. Includes 32 registers for storing data & transcendental functions. V1.0, S By: Steve Bonner
Mach	"mouse accelerator" prog. with hotkeys, features of sun mouse, clicktofront, and popcli, a title bar clock with a bbs online charge accumulator, etc.V1.6a, S By: Brian Moats
PatEdit	A pattern editor for creating patterns to input to the Amiga SetAfPt macro call. Call sets the area fill pattern for the area filling graphics (RectFill, AreaDraw, etc). Includes source. By Don Hyde
QMan	Mandelbrot generator written partially in assem. for speed. Includes source. By Steve Bonner

Fred Fish Disk 131

Dfc	Copies disks like Maurader, but multitasks. Replaces diskcopy and format (smaller than either). Intuition interface. S By: Tomas Rokicki
HyperBase	Shareware database management system. V1.6, Binary only, source available from authors. FF58 update. By: Michael MacKenzie, Marc Mengel, & Craig Norborg
Life	A new version of Tomas's ancient Life game, with a new macro language for setting up patterns, good examples, S By: Tomas Rokicki
Mackie	A Popcli replacement that draws pretty lines on the screen in blanking mode. Includes source. Author: Software Distillery; enhancements by Tomas Rokicki
Mg1b	A version of Mg1b with an ARexx port and other improvements by Tomas Rokicki. Define macros & bind them to function keys in startup file. Includes source. Author: Various; enhancements by Rokicki
WFrags	Another version of Frags, Pops up a little window that updates occasionally. Good for developers to monitor what progs. are doing to memory. S By: Tomas Rokicki

Fred Fish Disk 132

Berserk	Animation, a "must see" for every Amiga user, and ranks with "Juggler" as a premier demo for the Amiga. The difference between this distribution, and FF100, this one includes "source". use it as an example for creating animations. Fred Fish felt it was appropriate to have at least one animation that was available at the "source code" level. Author: Leo Schwab

Fred Fish Disk 133

Conman	Shareware replacement for the standard console handler, provides line editing and command line histories completely transparent to any application program that uses CON: windows. V1.1, binary only, updateoFF100. New features include additional editing keys, fast search keys, undo key, clear history command, and more. Author: William Hawes
Crc	Two programs useful for generating 16-bit CRC listings of the contents of disks, and verifying that a given disk's files still compute to the same CRC's as listed. V1.0, binary only. By: Don Kindred
CrcLists	Complete CRC check files for FF1-128 using the Crc program included on this disk. These were made directly from Fred's master disks. Author: Fred Fish
Overscan	Patches the Intuition library so that sizable windows with MaxHeight of 200 (400 in interlace) and screens with Height of 200 (400 in interlace) will take advantage of the PAL overscan capability of Intuition V1.2. Useful only for European users who wish to run software written for the US market, without modifying the applications, but still using the additional space. S By Ari Freund

Fred Fish Disk 134

BoingThrows	50 frame HAM animation done with Sculpt-3D, and DigiPaint. The animation took about 325 hours of runtime to generate. By Marvin Landis
Browser	Workbench tool, using text-only windows, makes all files in the system accessible for executing, copying, moving, renaming, deleting, etc. Billed as a "programmers workbench". V1.2, binary only. BY Peter da Silva
Dme	V1.29 of Matt's text editor. Simple WYSIWYG editor designed for programmers. Arbitrary key mapping, fast scrolling, title-line statistics multiple windows, & ability to iconify windows. FF113 update,S. By Matt Dillon

For PDS orders, please use forrn on page 112

Find	Utility searches for files that satisfy a given boolean expression of attributes, starting from a root pathname and searching recursively down through the hierarchy of the file system. Like the Unix find program. V1.0, includes source. By Rodney Lewis
Library	Demo version of a shareware program that stores textual information without regard to structure or content, and allows complicated searching for specific patterns. , B only. By: Bill Brownson
SmartIcon	Shareware Intuition objects iconifier. V1.0 is limited to iconifying windows, adds a new "iconify gadget" to each window, when clicked, iconifies the window into an icon in the ram: disk. B only, source available from author. By: Gauthier Groult

Fred Fish Disk 135

TeXF	A selection of 78 TeX fonts, with a conversion program to convert them to Amiga fonts. 22 different fonts at various sizes, ranging from 15 pixels high to more than 150 pixels. Conversion program can also be used with the fonts distributed with AmigaTeX, yielding an additional 1000+ fonts for use withother Amiga programs. V2.5, binary only. By: Ali Ozer

Fred Fish Disk 136

AsmToolBox	Assembler "toolbox" created to make interfacing between assembler programs and AmigaDOS easy. With source. By Warren Ring
Bison	A replacement for unix "yacc" command. From the GNU (GNU is Not Unix) effort. Port of the latest GNU version, by William Loftus, with the goal of preserving all of bison's current features. Includes source & test pro. "calc". By: Bob Corbett and Rich& Stallman.
Iff2Pcs	Interactive puzzle progr. takes any IFF file containing up to 16 colors, and breaks it into squares to make a puzzle the user can then piece back together again. V1.1, update of FF122, includes source. By Ali Ozer
Paste	Version of the Unix paste utility. Paste concatenates corresponding lines of the specified files into a single output line (horizontal or parallel merging) or concatenates them into alternate lines (vertical or serial merging). S. By: David Ihnat
YaBoingII	Game prog. demonstrating hardware sprite usage, including collision detection. Update of FF36. S. By: Ali Ozer, based on original by Leo Schwab
Zoo	File archiver, like "arc" in concept, but different in implementation and user interface details. Includes features that "arc" lacks (such as file/path names up to 255 characters in length). V 1.71, update of FF108. B. By: Rahul Dhesi, Amiga port by Brian Waters

Fred Fish Disk 137

Ct	Program to display images from a CT scanner, along with several interesting sample images of scans of real people, including a skull, brain, heart, and spine. Each image is 256 by 256 pixels in 2048 gray scale. The display software, though it has a primitive user interface, is quite powerful, including functions like convolutions, averaging, laplacians, unsharp masking, edge detection, gradients, etc. Binary only. Author: Jonathan Harman
JeansIcons	Miscellaneous cute icons created for AMUC's monthly newsletter disk. Submitted by Stephen Vermeulen. Author: Steve Jeans
Muncho	A cute little program which plays a digitized sound sample when you insert or remove a disk from your drive. If you don't like the sounds, you can replace them with your own. Binary only. By: Andrew Werth
Sit	Update to the Set Icon Type prog. on FF107. V1.10, includes source. Author: Stephen Vermeulen
VGad	A new gadget editor that takes two pictures of the window and its gadgets, one being the normal gadget state and the other being the fully selected state, then merges the data and converts to C source code. V1.0, binary only. Author: Stephen Vermeulen
VirusX	A boot sector virus check program that runs in the background and automatically checks all inserted disks for a nonstandard boot sector. Such disks can optionally have their boot sector rewritten to remove the virus. Includes source. Author: Steve Tibbett
VLabel	Program to print fancy customized disk labels. Combines an IFF picture and up to 50 lines of text (which may be placed arbitrarily in any font or point size) then print the result. The IFF picture can be virtually any size (up to 1008 by 1000). It will also print labels from a batch file produced by SuperBase. V1.20, binary only. By: Stephen Vermeulen

Fred Fish Disk 138

AmigaLine	A series of various technical notes for Amiga programmers. By: Byrce Nesbitt
Diff	Uses the same algorithm as the Unix diff program and also produces context diffs, suitable for use with patch. Binary only. By: Unknown (Decus C diff?)
Foreach	A simple but useful program that expands a wild card file specification and then invokes the specified command once per expanded filename, with the expanded filename as the command argument. Includes source. Author: Jonas Flygare
MacFont	A conversion tool to convert Mac fonts to Amiga fonts. Binary only. By: John O'Neill and Rico Mariani
ModulaTools	Various useful routines for those using in Modula on the Amiga. Update to FF94, S . By: Jerry Mack
Vt100	Two new versions of Dave's vt100 terminal emulator. One version, based on vt100 2.6, has been enhanced by John Barshinger to include an iconify feature, add full 132 column support using overscan, and other features (binary only). The second version is release 2.8 of the main-stream version of vt100, as enhanced and supported by Tony Sumrall. S. By: Dave Wecker

Fred Fish Disk 139

AmiCron	An enhanced and debugged version of AmiCron 2.3 from FF113. Includes source. By: Steve Sampson, Rich Schaeffer, Christian Balzer
ListScanner	A nice little utility to display all the Exec lists. Simular to Xplor utility FF73. Includes source in assembler. By: Heiko Rath
ProCalc	Simulates HP-11C programmable calculator. Both English & German versions. Shareware, B only. By: Gotz Muller
RemLib	Removes a specified library (if currently unused) or displays some info on all available libraries. Source in assembler. By: Heiko Rath
TurboBackup	A fast mass floppy disk duplicator with enforced verify mode to prevent errors. V1.0, binary only. By: Steffen Stempel and Martin Kopp
WArranger	Sends a window, identified by its name, to the front or back, without selecting it. Useful with AmiCron. Works on all screens. Includes source in assembler. By: Heiko Rath
WheelChairSim	A wheelchair simulator developed as a project for the Technical Resource Centre and the Albert Children's Hospital, to allow the matching of a wheelchair joystick to a child's handicap and allow the child to practice using the chair in a safe (simulated) environment. Binary only. Author: Unknown, submitted by Dr. Mike Smith

Fred Fish Disk 140

SBProlog	Volume 1 of the 2 volume Stony Brook Prolog (SBP) distribution, V2.3.2. This volume contains the executables and libraries. Volume 2, on FF141, contains the C and Prolog Source. By: Logic Programming Group at SUNY, Stony Brook Amiga port by David Roch & Scott Evernden

Fred Fish Disk 141

SBProlog	Volume 2 of the 2 volume Stony Brook Prolog (SBP) distribution, version 2.3.2. Volume 2 contains the C and Prolog source code. Volume 1, on FF140. By: Logic Programming Group at SUNY, Stony Brook Amiga port by David Roch and Scott Evernden
SmallC	An Amiga port of the Small-C compiler, written by Ron Cain and published in Dr. Dobb's Journal, in about 1980. Small-C is a rather small subset of the full 'C' language. It is capable of compiling itself, and other small, useful programs. Requires an assembler and linker to complete the package and produce working executables. Source and binary. By: Ron Cain. Amiga port by Willi Kusche.

Fred Fish Disk 142

Diff	Program uses same algorithm as Unix diff prog. and produces context diffs, suitable for use with patch.Same as FF138, but now includes the missing files (including source code). Author: Unknown (Decus C diff)
FracGen	Generates fractal pictures from "seeds" you create. Unlike any of the other "fractal generators", it can be used to load and display previously created fractal pics, modify existing fractals, or create your own fractals. V1.1, B. By:Doug Houck
SciSubr	Scientific Subroutine Package from DECUS, ported to the Amiga to run with Absoft Fortran. A valuable resource of mathematical and statistical source code for those doing Fortran work on the Amiga. Author: Unknown; ported to the Amiga by Glenn Everhart

Fred Fish Disk 143

Rim	RIM-5 (Relational Information Manager), a full relational DBMS suitable for VERY large databases using B-Tree data storage, crude (by today's standards) user interface, but full source code is provided. RIM runs on a wide variety of systems, small and large, and produce compatible databases. Includes a built in HELP database and a programming language. Full Fortran source code and documentation included. Author: Various, Amiga port by Glenn Everhart

Fred Fish Disk 144

AnalytiCalc	V22-3D of Glenn Everhart's large and powerful spreadsheet program, update to FF104. Extra features "to have some pretentions of acting as an 'integrated system'. A virtual memory system supporting up to18000 columns and 18000 rows, multiple equations per cell, an outlining system, built-in cell annotation, and datafile access from any cell(s) of the sheet, plus an array of functions not present in most commercial spreadsheets. Source and documentation in arc'd form.

Fred Fish Disk 145

Csh	Modification of csh like shell to provide file name completion and argument execution. Requires ARP 1.1. Binary only, but includes diffs for the reference 2.07 source base. Author: Matt Dillon; enhancements by Johan Widen
DMouse	Versatile screen blanker, mouse blanker, auto window activator, mouse accelerator, popcli style programmable command key, pop window to front, push window to back, etc, widget. Very useful program! . V1.06, includes source. Author: Matt Dillon
Net	Link protoco provides essentially an unlimited number of reliable connections between processes on two machines, where each can be either an Amiga or a Unix (BSD4.3) machine. Works on the Amiga with any EXEC device that looks like the serial.device. Works on UNIX with tty and socket devices. Achieves better than 95% average throughput on file transfers. V1.20,-includes sources for both the Amiga and Unix versions. Author: Matt Dillon
Tab	Tablature writing program, with intruments for a banjo and string guitar. Binary only. Author: Jeff deRienzo
TinyProlog	VT-PROLOG is a simple prolog interpreter provided with full source code to encourage experimentation with the PROLOG language and implementations. Version 1.1, includes source. Author: Bill and Bev Thompson

Fred Fish Disk 146

Blanker2	A screen blanking program that turns the screen black after 90 seconds of keyboard and mouse inactivity. V1.27.88, includes source. Author: Joe Hitchens
C-Light	A demo copy of a commercial ray tracing program, identical to commercial version but limited to ten objects per scene. Binary only. Author: Ronald Peterson
CrcLists	Complete CRC check files for FF129-141 and FF143-145 of the library, using the crc program from FF133. Made directly from Fred's master library. FF142 omitted due to a problem with the crc program. Author: Fred Fish
DmeMacros	A set of DME macros which utilize templates to turn DME into a language-sensitive editor for C, Pascal, Modula-2, and Fortran. By Jerry Mack
MemoPad	A shareware intuition-based memo reminder program. Nicely done. V1.1, binary only. Author: Michael Griebling

Fred Fish Disk 147

MicroGNUEmacs	MicroGNUEmacs(MG 2b) contains many additions and enhancements since the original works by Dave Conroy (credit belongs to all contributors and Beta testers. Note: Amiga specific source code files and the document files have been archived. An executable copy of the PDS archive program "Zoo" is in the "c" directory

Fred Fish Disk 148

EFJ	"Escape from Jovi" A machine-code game featuring hi-res scrolling, large playfield, disk-based Hi-Score list, stereo sound, and multiple levels. Use a joystick in port 2 to control the ship. B, shareware ($8). By: Oliver Wagner
Fme	Nicely done map editor for the Fire-Power (tm) game. Features interlaced hi-res with intuition interface. See the "Readme.fnf" file for information on making a bootable disk. Includes source. Author: Gregory MacKay
HandyIcons	Adds a menustrip to the WorkBench window that allows you to run selected Workbench Tools by menu selection. Can be set up to provide custom environments. Current version supports only WorkBench Tools and not Projects. Binary by: Alan Rubright
Scrambler	A simple program that will encode/decode a text file into illegible gibberish, which resembles executable code, to evade prying eyes. Version 0.01, Binary only. Author: Foster Hall

Fred Fish Disk 149

AnimalSounds	A sample of digitized animal sounds along with a simple sound player. Authors: The Trumor Company, Inc. Sound Player by Don Pitts
DX-VoiceSorter	Written to be used with Jack Deckard's VoiceFiler program. (Disk 82). It allows for the sorting of a number of voicefiles stored using that program into a new voicefile of voices made up from various files. Includes source. Author: David Bouckley
Keep	A nice little utility program with an intuition interface for BBS and network junkies who download messages in one large file and then read them off-line. Using only the mouse, you can drive through such files a message at a time, examine each at your leisure and tag those you wish to keep. Version 1.2, binary only, but source available with donation to author. Author: Tim Grantham
Less	Like Unix "more", only better, with forward and backward scrolling, searching and positioning by percent of file and line number, etc. Now lets you also print the current file. Very useful! This is Amiga version 1.3, an update to the version on disk number 92. Includes source. Author: Mark Nudelman, Amiga port by Bob Leivian
Scheme	"Scheme is a statically scoped and properly tail-recursive dialect of the Lisp programming language invented by Guy Lewis Steele Jr. and Gerald Jay Sussman." Binary only. Amiga port by Ed Puckett

Fred Fish Disk 150

AirFoil	An update to the Airfoil generator on disk #71. Generates airfoil models as well as their corresponding streamline and pressure distributions. Includes source. Authors: Russell Leighton Addendum by David Foster
DC10	An AmigaBasic DC-10 instrument flight simulator. Appears to be quite in-depth with flight-planning and take-off options along with an extensive documentation file. Requires rebuilding on a separate disk and was successfully done so by following the author's instructions in the ReadMe_First file. Author: Jan Arkesteijn
ExecLib	A working example of how to build and use user-defined disk-resident libraries. Of special interest to developers working with Lattice C. Author: Alex Livshits
Iconizer	A utility program that saves your current mouse pointer to a small icon. You can restore the pointer just by double-clicking on its icon. Allows for building a whole library of pointers and to use them whenever you want. Binary only. Author: Alex Livshits
Pilot	An implementation of the PILOT language for the Amiga, including a demo done for the National Park Service. PILOT is a limited use language for use in educational and computer based instruction programs. Binary only with Beta test kit available from authors. Author: Terry LaGrone
StealMemBoot	A small utility designed to be a direct replacement for NoFastMem kind of programs. It modifies the boot block of a disk, so when you boot with it, all memory allocations will return only CHIP memory. Author: Alex Livshits

Fred Fish Disk 151

GlobeDemo	A graphics demo which displays very smooth transitions of the rotating earth. Features a pop-up menu. Includes source. Author: Bob Corwin
Icons	Yet another potpourri of interesting icons to choose from if you need one for your own program. Author: Dave Turnock
Pcopy	A small intuition-based disk copier similiar to the resident "DiskCopy" except with write-verify and other user-selectable options. Useful for making multiple copies with reliable data. Requires two disk drives. Includes source. Author: Dirk Reisig
SCT	A CLI-based utility (SetColorTable) for displaying and/or setting a screen's colors. You can save the colors of a screen to be restored later, or copy one screen's colors to another. Includes source. Author: aklevin
SlideShow	Very nicely done slide-show program written in assembly language. Features forward/backward presentation and creative screen wipes. Currently works only with IFF lo-res pictures. Executable only along with some new IFF pictures to have come my way. Shareware ($16). Authors: Mike McKittrick and Sheldon Templeton
Surveyor	A little utility that opens a window on the current screen and displays information about the pointer. Allows for absolute or relative measurement between two points on the screen. Very handy for precise positioning of icons and such. Includes source. Author: Dirk Reisig

Fred Fish Disk 152

Blk	A requester making tool employing various recursive algorithms including a recursive parser. It takes input text files and converts them to C-source for including as requester declarations. Includes source. Author: Stuart Ferguson
RunBack	A variant of Rob Peck's RunBackGround program from disk number 73. Allows you to start a new CLI program and run it in the background, then closes the new CLI. This version automatically searches the command-search-path to find the program. Includes source. By Daniel Barrett
UUCP	This is a version of uucp (Unix to Unix Copy Program) for the Amiga, along with some miscellaneous support utilities like cron, mail, and compress. Includes source. Author: Various, submitted by William Loftus

Fred Fish Disk 153

Dme	Version 1.30 of Matt's text editor. Dme is a simple WYSIWYG editor designed for programmers. It is not a WYSIWYG word processor in the traditional sense. Features include arbitrary key mapping, fast scrolling, title-line statistics multiple windows, and ability to iconify windows. Update to version on disk number 134, includes source. Author: Matt Dillon
HP11	Emulates an HP11C calculator including the program mode. Features an ON/OFF button that turns the calculator into an icon that will sit and wait until you need it again. Documentation on the features is scarce, perhaps some industrious HP owner could write a small tutorial for the benefit of those that don't own an HP calculator. Binary only. Author: David Gay
HPMam	A program to manipulate settings and fonts on HP LaserJet+ printers and compatibles. Includes an Intuition interface and some sample picture files. Version 1.0, binary only, shareware. Author: Steve Robb
Synthemania	An interesting, very small (and very persistent!) musical piece. If you plan on stopping it without using three fingers, you better read the document file first! Binary only. Author: Holger Lubitz

Fred Fish Disk 154

Ada	An Ada Syntax checker for the amiga. Includes lex and yacc source. Author: Herman Fischer; updates by William Loftus
AssemblyDemos	A interesting group of assembly language demos for your visual and aural pleasure. Binary only. Author: Foster Hall
DiskLib	Two utilities for those people who like to split up PD disks into disks of different categories. Includes source. Author: Wilson Snyder
Guardian	Another virus diagnosing and vaccination program. Recognizes any non-standard bootblock. Includes a small utility program to permanently place the program on a copy of your kickstart disk in place of the seldom (if ever!) used Debug() function. Binary only. Author: Leonardo Fei
PrintSpool	A print-spooling program. Very useful for printing files in the background. Many command-line options. Version 1.0.0, Includes source. Author: Frangois Gagnon
Utilities	A group of four little utility programs: UnDelete - Undelete a file from floppy (DF0:) to any device you request, checks for a disk in the drive and allows you to abort cleanly with a CTRL 'C'. WhereIs - Looks for a file and or directory defaults to the current device CAL - Clone of the Unix CAL command, dates from the year 1 to 9999. DClock Simple title bar clock/memory gauge with pop to front.
VirusX	An update to the virus-detecting program of the same name on disk number 137. This version also checks for the Byte-Bandit strain. Version 1.21, Includes source. Author: Steve Tibbett
Virus_Alert!	Yet another anti-virus program with a twist. Once installed a message is displayed just after a warm or cold boot notifying the the user that the disk and memory are virus-free, and forcing a mouse-button press before continuing. Anything writing to the bootblock thereafter will destroy the message and a normal virus- infected boot (???) will take-place. Versions 1.01 and 2.01, Binary only. Author: Foster Hall
Wicon	A "Window Iconifier". Allows you to turn your windows into small icons which can be later recalled. Currently installed with MacWin to give your windows a "rubber-banding" effect. Version 1.14, Includes source. Author: Steven Sweeting Introducing the Amiga

To Be Continued......

In Conclusion

To the best of our knowledge, the materials in this library are freely distributable. This means they were either publicly posted and placed in the public domain by their authors, or they have restrictions published in their files to which we have adhered. If you become aware of any violation of the authors' wishes, please contact us by mail.

IMPORTANT NOTICE!

This list is compiled and published as a service to the Commodore Amiga community for informational purposes only. Its use is restricted to non-commercial groups only! Any duplication for commercial purposes is strictly forbidden. As a part of Amazing Computing™, this list is inherently copyrighted. Any infringement on this proprietary copyright without expressed written permission of the publishers will incur the full force of legal actions.

Any non-commercial Amiga user group wishing to duplicate this list should contact:

PiM Publications, Inc.
P.O.Box 869
Fall River, MA 02722

PiM Publications Inc. is extremely interested in helping any Amiga user groups in non-commercial support for the Amiga. •AC•

Amaze Me

Please use this order form when subscribing to Amazing Computing™, ordering Back issues, or ordering Amiga™ Public Domain Software

Name______________________________
Street______________________________
City ________________ **St.**____ **Zip** __________
Amount Enclosed __________

Please circle the appropriate item: **New Subscription** **Renewal**

Please start my subscription to Amazing Computing™ with the next available issue or renew my current subscription.

I have enclosed $24.00 for 12 issues in the U.S. ($36.00 Canada and Mexico, $44.00 foreign surface).
All funds must be in U.S. Currency on a U.S. Bank

Back Issues:
$5.00 each US, $6.00 each Canada and Mexico, $7.00 each Foreign Surface.

Please circle your Back issue choices below:

Vol1.1 Vol1.2 Vol1.3 Vol1.4 Vol1.5 Vol1.6 Vol1.7 Vol1.8 Vol1.9 Vol2.1 Vol2.2 Vol2.3 Vol2.4 Vol2.5
Vol2.6 Vol2.7 Vol2.8 Vol2.9 Vol2.10 Vol2.11 Vol2.12 Vol3.1 Vol3.2 Vol3.3 Vol3.4 Vol3.5 Vol3.6 Vol3.7
Vol3.8 Vol3.9 Vol3.10

Public Domain Software:

$6.00 each for subscribers (yes, even the new ones!)
$7.00 each for non subscribers
(three disk minimum on all foreign orders)

Please circle your Public Domain Software choices below:

Amazing on Disk:

A#1...Source & Listings V3.9

Amicus:

A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 A13
A14 A15 A16 A17 A18 A19 A20 A21 A22 A23 A24 A25 A26

Fred Fish:

FF1 FF2 FF3 FF4 FF5 FF6 FF7 FF8 FF9 FF10 FF11 FF12 FF13
FF14 FF15 FF16 FF17 FF18 FF19 FF20 FF21 FF22 FF23 FF24 FF25 FF26
FF27 FF28 FF29 FF30 FF31 FF32 FF33 FF34 FF35 FF36 FF37 FF38 FF39
FF40 FF41 FF42 FF43 FF44 FF45 FF46 FF47 FF48 FF49 FF50 FF51 FF52
FF53 FF54 FF55 FF56 FFNA FF58 FF59 FF60 FF61 FF62 FF63 FF64 FF65
FF66 FF67 FF68 FF69 FF70 FF71 FF72 FF73 FF74 FF75 FF76 FF77 FF78
FF79 FFNA FF81 FF82 FF83 FF84 FF85 FF86 FF87 FFNA FF89 FF90 FF91
FF92 FF93 FF94 FF95 FF96 FF97 FF98 FF99 FF100 FF101 FF102 FF103 FF104
FF105 FF106 FF107 FF108 FF109 FF110 FF111 FF112 FF113 FF114 FF115 FF116 FF117
FF118 FF119 FF120 FF121 FF122 FF123 FF124 FF125 FF126 FF127 FF128 FF129 FF130
FF131 FF132 FF133 FF134 FF135 FF136 FF137 FF138 FF139 FF140 FF141 FF142 FF143
FF144 FF145 FF146 FF147 FF148 FF149 FF150 FF151 FF152 FF153 FF154

(NA denotes disks removed from the collection)

Please complete this form and mail with check or money order to:

PiM Publications, Inc.
P.O.Box 869
Fall River, MA 02722

Please allow 4 to 6 weeks for delivery

8ColorWB	Modifies the Workbench so three bitplanes are used, icons can have eight colors, instead of four, eight-color icons are included. Public domain program "zapicon" or "brush2icon" converts eight-color IFF brushes to icons, to use Deluxe Paint to make icons for this new Workbench.
BrushIcon	Converts brushes to icons (bizarr docs).
Egraph	Graphing prog reads [x,y] values from a file and displays them on the screen, similar to the same-named Unix program.
Keep 1.1	Message-managing program for telecommunications, lets you save messages from an online transcript to another file, understands the message format of the national networks and several types of bulletin board software. Moves through the transcript and save messages
Kill.fastdir	Speed up directory access, it creates a small file in each directory on a disk which contains the information about the files, will also remove all the "fastdir" files from each directory. by CLImate's authors

The LaceWB program changes between interlace and non-interlace Workbench. Previously, you were forced to reboot after changing Preferences to an interlaced screen. This program flips between the normal and extended screen heights.

PW_Utility	A shareware utility for ProWrite users, changes margin settings and font types.
Guru	A CLI program, prints out probable causes for Guru mediations; C source included.
DiskWipe	Latest from Software Distillery, removes files from directories or disk drives, much faster than "delete."
Snow	AmigaBasic makes snowflake designs.
Mlist	Mailing list database.
Softballstats	Maintain softball statistics/ team records.
Dodge	Short Modula-2 program moves the Workbench screen around after a period of time, prevents monitor burn-in.

AMICUS Disk 26

Todor Fay's SoundScape module code from his Amazing Computing articles. The source to Echo, Chord, TX, and VU is included. The Lattice and Manx C source code is here, along with the executable modules.

ImageMaker	Interesting tool edits Image structures for C, loads & saves C code directly.
Claz2	Update of prog to convert IFF images to PostScript files for printing on laser printers
SDBackup	Hard disk backup prog with Lempel-Ziv compression to reduce the necessary number of disks.
TCB	Prints information about tasks and processes in the system; assembler source is included.
FunBut	Lets a function key act like a rapid series of left mouse button events.
DC	A handy program for people who use an Amiga 1020 5 1/4 inch drive as an AmigaDOS floppy. A Workbench program that sends a DiskChange signal to the operating system: instead of typing "diskchange df2:" over and over again, just click on the icon. C source included.
System config	File makes screen 80 columns wide of text in the Scribble! word processor.
Dick2Ram	2 programs to move the Scribble! spelling dictionary to and from the RAM disk.
Lexical	Analyzes a text file and gives the Gunning-Fog, Flesch, and Kincaid indices which measure readability.
HexDump	Modula-2 program to display memory locations in hexadecimal.
Tartan	AmigaBasic; design Tartan plaids.
DirMaster	Disk catalog program.
BMP	plays 8SVX sampled sounds in the background while something else is happening in the Amiga, as your Amiga is booting, for example.
ShowPt	CLI program changes your pointer to a given pointer.

AMICUS 26 also has a collection of mouse pointers, & Workbench program to display them

Fred Fish Public Domain Software

Fred Fish Disk 1:

amigademo	Graphical benchmark for comparing amigas.
amigaterm	simple communications program with Xmodem
balls	simulation of the "kinetic thingy" with balls on strings
colorful	Shows off use of hold-and-modify mode.
dhrystone	Dhrystone benchmark program.
dotty	Source to the "dotty window" demo on the Workbench disk.
freedraw	A small "paint" type program with lines, boxes, etc.
gad	John Draper's Gadget tutorial program
gfxmem	Graphical memory usage display prog.
halfbrite	demonstrates "Extra-Half-Brite" mode, if you have it
hello	simple window demo
latffp	accessing the Motorola Fast Floating Point library from C
palette	Sample prog. to design color palettes.
trackdisk	Demonstrates use of the trackdisk driver.
requesters	John Draper's requester tutorial and example program.
speech	Sample speech demo program. Stripped down "speechtoy".
speechtoy	Another speech demo program.

Fred Fish Disk 2:

alib	Object module librarian.
cc	Unix-like frontend for Lattice C compiler.
dbug	Macro based C debugging package. Machine independent.
make	Subset of Unix make command.
make2	Another make subset command.
microemacs	Small version of emacs editor, with macros, no extensions
portar	Portable file archiver.
xrf	DECUS C cross reference utility.

Fred Fish Disk 3:

gothic	Gothic font banner printer.
roff	A "roff" type text formatter.
ff	A very fast text formatter
cforth	A highly portable forth implementation. Lots of goodies.
xlisp	Xlisp 1.4, not working correctly.

Fred Fish Disk 4:

banner	Prints horizontal banner
bgrep	A Boyer-Moore grep-like utility
bison	CNU Unix replacement 'yacc', not working.
bm	Another Boyer-Moore grep-like utility
grep	DECUS grep
kermit	simple portable Kermit with no connect mode.
MyCLI	Replacement CLI for the Amiga. V. 1.0
mandel	A Mandelbrot set program, by Robert French and RJ Mical

Fred Fish Disk 5

cons	Console device demo program with supporting macro routines.
freemap	Creates a visual diagram of free memory
input.dev	sample input handler, t raps'key or mouse events
joystick	Shows how to set up the gameport device as a joystick.
keyboard	demonstrates direct communications with the keyboard.
layers	Shows use of the layers library
mandelbrot	IFF Mandelbrot program
mouse	hooks up mouse to right joystick port
one.window	console window demo
parallel	Demonstrates access to the parallel port.
printer	opening and using the printer, does a screen dump, not working
print.support	Printer support routines, not working.
proctest	sample process creation code, not working
region	demos split drawing regions
samplefont	sample font with info on creating your own
serial	Demos the serial port
singlePlayfield	Creates 320 x 200 playfield
speechtoy	latest version of cute speech demo
speech.demo	simplified version of speechtoy, with IO requests
text.demo	displays available fonts
timer	demos timer.device use
trackdisk	demos trakcdisk driver

Fred Fish Disk 6:

compress	like Unix compress, a file squeezer
dadc	analog clock impersonator
microemacs	upgraded version of microemacs from disk 2
mult	removes multiple occuring lines in files
scales	demos using sound and audio functions
setparallel	Allows changing parallel port parameters
setserial	Allows changing serial port parameters.
sortc	quicksort based sort program, in C
stripc	Strips comments and extra whitespace from C source

Fred Fish Disk 7:

This disk contains the executables of the game Hack V 1.0.1.

Fred Fish Disk 8:

This disk contains the C source to Hack on disk 7.

Fred Fish Disk 9:

moire	Draws moire patterns in black and white
MVP-FORTH	Mountain View Press Forth, version 1.00.03A. A shareware version of FORTH from Fantasia Systems.
proff	a more powerful text formatting program
setlace	Prog to toggle interlace mode on and off
skewb	a rubic's cube type demo
sparks	moving snake Graphics demo

Fred Fish Disk 10:

conquest	An interstellar adventure simulation game
dehex	convert a hex file to binary
filezap	Patch program for any type of file.
fixobj	Strip garbage off Xmodem transferred files.
iff	Routines to read and write iff format files.
ld	simple directory program
ls	Minimal UNIX ls, with Unix-style wildcarding, in C
sq,usq	file squeeze and unsqueeze
trek73	Star Trek game
yachtc	Dice game.

Fred Fish Disk 11:

dpslide	slide show program for displaying IFF images with miscellaneous pictures

Fred Fish Disk 12:

amiga3d	Shows a rotating 3 dimensional solid "Amiga sign".
ArgoTerm	a terminal emulator program, written in assembler
arrow3d	Shows a rotating 3 dimensional wire frame arrow.
ld4	directory listing program
IconExec	
SetWindow	two progs for launching progs from Workbench, presently only works under CLI.
SetAlternate	Makes an icon show a second image when clicked once
StarTerm	terminal emulator, with ASCII Xmodem, dialer, more.

Fred Fish Disk 13:

A Bundle of Basic programs, including:

Jpad	toybox	ezspeak	mandlebrot
xmodem	3dsolids	addbook	algebra
ror	amgseq1	amiga-copy	band
bounce	box	brickout	canvas
cardfi	circle	colorcircles	Copy
cubes1	cutpaste	datedogstar	
dragon	draw	dynamictriangle	
Eliza	ezterm	fillibuster	fractal
fscape	gomoku	dart	haiku
hal9000	halley	hauntedM	hidden
join	loz	mandel	menu
minipaint	mouse	Orthello	patch
pena	pinwheel	gboxrandom-circles	
Readme	rgb	rgbtest	Rord
sabotage	salestalk	shades	shapes
shuttle			
sketchpad	spaceart	speakspeach	
speecheasy	spell	sphere	
spiral	striper	superpad	suprshr
talk	terminal		
termtest	tom	topography	triangle
wheels	xenos	xmostriper	

(note: some programs are Abasic, most are Amigabasic, and some programs are presented in both languages)

Fred Fish Disk 14:

amiga3d	update of #12, includes C source to a full hidden surface removal and 3D graphics
beep	Source for a function that generates a beep sound
dex	extracts text from within C source files
dimensions	demonstrates N dimensional graphics
filezap	update of disk 10, a file patch utility
gfxmem	update of disk 1, graphic memory usage indicator
gi	converts IFF brush files to Image struct, in C text.
pdterm	simple ANSI VT100 terminal emulator, in 80 x 25 screen
shell	simple Unix 'csh' style shell
termcap	mostly Unix compatible 'termcap' implementation.

Fred Fish Disk 15:

Blobs	graphics demo, like Unix 'worms'
Clock	simple digital clock program for the title bar
Dazzle	An eight-fold symmetry dazzler program. Really pretty!
Fish	double buffered sequence cycle animation of a fish
Monopoly	A really nice monopoly game written in AbasiC.
OkidataDump	Okidata ML92 driver and WorkBench screen dump program.
Polydraw	A drawing program written in AbasiC.
Polyfractals	A fractal program written in AbasiC.

Fred Fish Disk 16:

A complete copy of the latest developer IFF disk

Fred Fish Disk 17:

The NewTek Digi-View video digitizer HAM demo disk

Fred Fish Disk 18:

AmigaDisplay	dumb terminal program with bell, selectable fonts
Ash	Prerelease C Shell-like shell program, history, loops, etc.
Browser	wanders a file tree, displays files, all with the mouse
MC68010	docs on upgrading your Amiga to use a MC68010
Multidim	rotate an N dimensional cube with a joystick
PigLatin	SAY command that talks in Pig Latin
Scrimper	Screen image printer
Xlisp1.6	source, docs, and execut for a Lisp interpret.

Fred Fish Disk 19:

BlackJack	text-oriented blackjack game
JayMinerSlides	Slides by Jay Miner, Amiga graphics chip designer, showing flowchart of the Amiga internals, in 640 x 400.
Keymap_Test	test program to test the key mapping routines
LockMon	Find unclosed file locks, for programs that don't clean up.

Fred Fish Disk 20:

AmigaToAtari	converts Amiga object code to Atari form
DiskSalv	program to recover files from a trashed AmigaDOS disk.
Hash	example of the AmigaDOS disk hashing function
Hd	Hex dump utility ala Computer Language magazine, April 86
MandelBrots	Mandelbrot contest winners
MultiTasking	Tutorial and examples for Exec level multitasking
Pack	strips whitespace from C source
PortHandler	sample Port-Handler program that performs. Shows BCPL environment
Random	Random number generator in assembly, f or C or assembler.
SetMouse2	sets the mouse port to right or left
SpeechTerm	terminal Emulator with speech capabilities, XModem
TxEd	Demo editor from Microsmith's Charlie Heath

Fred Fish Disk 21

This is a copy of Thomas Wilcox's Mandelbrot Set Explorer disk. Very good!

Fred Fish Disk 22

This disk contains two new "strains" of microemacs.

Lemacs	version 3.6 by Daniel Lawrence. For Unix V7, BSD 4.2, Amiga, MS-DOS, VMS. Uses Amiga function keys, status line, execute, startup files, more.
Pemacs	By Andy Poggio. New features include <ALT> keys as Meta keys, mouse support, higher priority, backup files, word wrap, function keys.

Fred Fish Disk 23

Disk of source for MicroEmacs, several versions for most popular operating systems on micros and mainframes. For people who want to port MicroEmacs to their favorite machine.

Fred Fish Disk 24:

Conques	interstaller adventure simulation game
Csh	update to shell on Disk 14, with built in commands,named variables substitution.
Modula-2	A pre-release version of the single pass Modula-2 compiler originally developed for MacIntosh at ETHZ. This code was transmitted to the AMIGA and is executed on the AMIGA with a special loader. Binary only.

Fred Fish Disk 25

Graphic Hack	A graphic version of the game on disks 7 and 8 This is the graphics-oriented Hack game by John Toebes. Only the executable is present.

Fred Fish Disk 26

UnHunk	Processes the Amiga "hunk" loadfiles. Collect code, data, and bss hunks together, allows individual specificatio of code, data, and bss origins, and generates binary file with format reminiscent of Unix "a.out" format. The output file can be easily processed by a separate program to produce Motorola "S-records" suitable for downloading to PROM programmer. By Eric Black.
C-kermit	Port of the Kermit file transfer program and server.
Ps	Display and set process priorities
Archx	Yet another program for bundling up text files and mailing or posting them as a single file unit.

Fred Fish Disk 27

ABdemos	Amiga Basic demos; Caroly Scheppner.
NewConvertFD	creates .bmaps from fd files.
BitPlanes	finds addresses of and writes to bitplanes of the screen's bitmap.
AboutBmaps	A tutorial on creation and use of bmaps.
LoadILBM	loads and displays IFF ILBM pics.
LoadACBM	loads and displays ACBM pics.
ScreenPrint	creates a demo screen and dumps it to a graphic printer.
Disassem	Simple 68000 disassembler. Reads standard Amiga object files and disassembles the code sections. Data sections are dumped in hex. The actual disassember routines are set up to be callable from a user prog so instructions in memory can be disassembled dynamically. By Bill Rogers.
DvorakKeymap	Example of a keymap structure for the Dvorak keyboard layout. Untested but included because assembly examples are few and far between. By Robert Burns
Hypocycloids	Spirograph, from Feb. 84 Byte.
LinesDemo	Example of proportional gadgets to scroll a SuperBitMap.
MemExpansion	Schematics and directions for building your own homebrew 1 Mb memory expansion, by Michael Fellinger.
SafeMalloc	Program to debug 'malloc()' calls
ScienceDemos	Convert Julian to solar and sidereal time, stellar positions and radial velocity epoch calculations and Galilean satellite plotter. By David Eagle.

Fred Fish Disk 28

ABasic games by David Addison: Backgammon , Cribbage, Milestone, and Othello

Cpp	DECUS 'cpp' C preprocessor, & a modified 'cc' that knows about the 'cpp', for Manx C.
Shar	Unix-compatible shell archiver, for packing files for travel.
SuperBitMap	Example of using a ScrollLayer, syncing SuperBitMaps for printing, and creating dummy RastPorts.

Fred Fish Disk 29

AegisDraw Demo	Demo program without save and no docs.
Animator Demo	Player for the Aegis Animator files
Cc	Unix-like front-end for Manx C
Enough	Tests for existence of system resources, files, and devices
Rubik	Animated Rubik's cube program
StringLib	
Vt100	VT-100 terminal emulator with Kermit and Xmodem protocols

Fred Fish Disk 30

Several shareware programs. The authors request a donation if you find their program useful, so they can write more software.

BBS	an Amiga Basic BBS by Ewan Grantham
FineArt	Amiga art
FontEditor	edit fonts, by Tim Robinson
MenuEditor	Create menus, save them as C source, by David Pehrson
StarTerm3.0	Very nice telecom by J. Nangano

(Fred Fish Disk#30 is free if requested when ordered with at least three other disks from the collection.)

Fred Fish Disk 31

Life	Life game, uses blitter to do 19.8 generations a second.
Mandelbrot	Version 3.0 of Robert French's program.
MxExample	Mutual exclusion gadget example.
RamSpeed	Measure relative RAM speed, chip and fast.
Set	Replacement for the Manx "set" command for environment variables,with improvements.
Tree	Draws a recursive tree, green leafy type, not files.
TxEd	Crippled demo version of Microsmith's text editor, TxEd.
VDraw	Full-featured drawing program by Stephen Vermeulen.
Xicon	Invokes CLI scripts from icon
Ticon	Displays text files from an icon.

Fred Fish Disk 32

Address	Extended address book, AmigaBASIC
Calendar	Calendar/diary program, AmigaBASIC
DosPlus1	First volume of CLI oriented developer tools
DosPlus2	2nd volume of CLI oriented developer tools.
Executables only:	
MacView	Views MacPaint pics in Amiga low or high res, no sample pictures, by Scott Evernden.
Puzzle	Simulation of puzzle with moving squares.
ShowHAM	View HAM pictures from CLI.
Solitaire	ABasiC games of Canfield and Klondike, from David Addison.
Spin3	Graphics demo of spinning cubes, double-buffered example.
Sword	Sword of Fallen Angel text adventure game written in Amiga Basic.
Trails	Leaves a trail behind mouse, in Modula-2

Fred Fish Disk 33

3dstars	3d version of the "stars" program below.
Bigmap	Low-level graphics example scrolls bitmap with ScrollVPort.
Dbuf.gels	Double-buffered animation example for BOBs and VSprites.
DiskMapper	Displays sector allocation of floppy disks.
MemView	View memory in real time, move with joystick.
Oing	Bouncing balls demo
Sproing	Oing, with sound effects.
ScreenDump	Dumps highest screen or window to the printer.
Sdb	Simple database program from a DECUS tape.
Stars	Star field demo, like Star Trek.
TermPlus	Terminal program with capture, library, function keys, Xmodem, CIS-B protocols.
Vt100	Version 2.0 of Dave Wecker's VT-100 emulator, with scripts & function

Fred Fish Disk 34

Alint	Support files for Gimpel's 'lint' syntax checker
Blink	PD 'alink' compaible linker,faster,better.
Browser	Updated to FF 18 'browser', in Manx, with scroll bars, bug fixes.
Btree	b-tree data structure examples
Btree2	Another version of 'btree'
Calendar	Appointment calendar with alarm.
Less	File viewer, searching, position by percent, line number.
NewFonts	Set of 28 new Amiga fonts from Bill Fischer
Pr	Background print utility, style options, wildcards.
Requester	Deluxe Paint-type file requester, with sample.

Fred Fish Disk 35

ASendPacket	C example of making asynchronous I/O calls to a DOS handler, written by C-A
ConsoleWindow	C example of getting the Intuition pointer a CON: or RAW: window, for 1.2, by C-A.
DirUtil	Walk the directory tree, do CLI operations from menus
DirUtil2	Another variant of Dirutil.
FileRequester	Lattice C file requester module, with demo driver, from Charlie Heath.
MacView	Views MacPaint pictures in Amiga low or high res, with sample pictures, by Scott Evernden.
Plop	Simple IFF reader program
PopCLI	Sidekick-style program invokes a new CLI, with automatic screen blanking.
QuickCopy	Devenport disk copiers duplicate copy-protected disks.
ScrollPf	Dual playfield example, from C-A, shows 400 x 300 x 2 bit plane playfield on a 320 x 200 x 2 plane deep playfield.
SendPacket	General purpose subroutine to send AmigaDos packets.
SpriteMaker	Sprite editor, can save work as C data structure. Shareware by Ray Larson.
Tracker	Converts any disk into files, for electronic transmission. Preserves entire file structure. Shareware by Brad Wilson.
TriClops 3-D	space invasion game, formerly commercial, now public domain. From Geodesic Publications.
Tsize	Print total size of all files in subdirectories.
Unifdef	C preprocessor to remove given #ifdef'd sections of a file, leaving the rest alone. By Dave Yost
Vttest	VT-100 emulation test program. Requires a Unix system.

Fred Fish Disk 36

Acp	Unix-like 'cp' copy program
Clock	Updated version of clock on disk 15.
Csh	Manx 'csh'-like CLI, history, variables, etc.
DietAid	Diet planning aid organizes recipes, calories
Echo	Improved 'echo' command with color, cursor addressing
FixHunk	Fixs programs to let them run in external memory.
Fm	Maps the sectors a file uses on the disk.
KickBench	Docs, program to make a single disk that works like a Kickstart and Workbench.
Lex	Computes Fog, Flesch, and Kincaid readability of text files.
TunnelVision	David Addison ABasic 3D maze perspective game.
Vc	Visicalc-like spreadsheet calculator program.
Vt100	Version 2.2 of Dave Wecker's telecom program
YaBoing	Oing! style game program shows sprite collision detects

Fred Fish Disk 37

This disk is a port of Timothy Budd's Little Smalltalk system, done by Bill Kinnersley at Washington State University.

Fred Fish Disk 38

CSquared	Sep 86 Sci American, Circle Squared algorithm
FixObj	Strips garbage off Xmodem transfered object files
Handler	AmigaDOS handler (device) example from C-A
Hp-10c	Mimics a HP-10C calculator, written in Modula-2
IFFEncode	Saves the screen as an IFF file
IffDump	Dumps info about an IFF file
Jsh	BDS C-like CLI shell
NewStat	STATUS-like program, shows priority, processes
Reversi	Game of Reversi, version 6.1
UUdecode	Translate binary files to text, Unix-like programs
Vdraw	Drawing program, version 1.14
VoiceFiler	DX MIDI synthesizer voice filer program
Window	Example of creating a D OS window on a custom screen

Fred Fish Disk 39

AnsiEcho	'echo', 'touch', 'list', 'cls' written in assembler.
Display	Displays HAM images from a ray-tracing program, with example pictures.
Driver	Example device driver source, acts like RAM: disk
Xlisp	XLisp 1.7, executable only

Fred Fish Disk 40

Ahost	Terminal emulator with Xmodem, Kermit and CIS B protocols, function keys, scripts, RLE graphics and conference mode.
AmigaMonitor	Dynamically displays the machine state, such as open files, active tasks, resources, device states, interrupts, libraries, ports, etc.
Arc	Popular file compression system, the standard for transitting files
AreaCode	Program that decodes area codes into state and locality.
Blink	'alink' replacement linker, version 6.5
Cosmo	An 'asteriods' clone.
Dg210	Data General D-210 Terminal emulator
DirUtil	Windowed DOS interface program, V 1.4
DOSHelper	Windowed AmigaDOS CLI help program
PagePrint	Prints text files with headers, page breaks, line numbers
PopCLI	Starts a new CLI with a single keystroke, from any program, With a screen-saver feature. Version 2, w/source.
SpriteEd	Sprite Editor edits two sprites at a time
X-Spell	Spelling checker allows edits to files

Fred Fish Disk 41

AmigaVenture	Create your own text adventure programs in AmigaBasic.
Csh	Version 2.03 of Dillon's C sh-like shell. Executable only
Dbug	Macro based C debugging packageupdate to FF #2
DualPlayField	example from CBM, update to Intuition manual
GetFile	Heath's file requester, with source
LatXref	Cross reference of Lattice 3.10 header files
Lines	Line drawing demo program
SetFont	Changes font used in a CLI window
Vt100	Version 2.3 of the VT-100 terminal program.

Fred Fish Disk 42

This disk contains an Amiga version of MicroGNUEmacs.

Fred Fish Disk 43

BasicBoing	AmigaBasic program demos page flipping of a 3D cube
Bbm	Demo copy of B.E.S.T. Business Management System.
BbsList	A list of Amiga Bulletin Board Systems
Cc	C compiler frontends for Manx and Lattice C
Copper	A hardware copper list disassembler
InstIFF	Converts Instruments demo sounds to IFF sampled sounds
PopColours	Adjust RGB colors of any screen
SpriteClock	Simple clock is displayed on a sprite above all screens
ST Emulator	Non-serious Atari ST emulator
WBrun	Lets Workbench programs be run from the CLI
Wild	Two Unix shell style wild card matching routines

Fred Fish Disk 44

Icons	Miscellaneous icons
NewIFF	New IFF material from CBM for sampled voice and music files
RayTracePics	The famous ray-tracing pictures, from FF#39, now converted to IFF HAM format for "much" faster viewing.
ViewILBM	Displays normal and HAM ILBM files

Fred Fish Disk 45

Clue	Clue board game
Make	Another 'make', with more features
Pictures	Miscellaneous pictures
Update	Updates older disk with newer files fromanother disk
WhereIs	Searches a disk for files of given name

Fred Fish Disk 46

Asm	Shareware 68010 macro assembler, ROM Kernal Manual compatible
CheckModem	'execute' file program detects presence of modem
Egad	Gadget editor from the Programmers Network
Jive	Transforms a file from English to Jive.
My.lib	A binary only copy of Ma tt's alternate runtime library. Author: Matt Dillon
ProffMacros	Subset Berkeley 'ms' and 'mm' macros for 'proff'
ValSpeak	Transforms a file from English to Valley Speak.

Fred Fish Disk 47

3D-Arm	Simulation of a robotic a rm, very good graphics, teaching tool, including C source.
Juggler	Eric Graham's stunning HAM animation of a robot juggler
VT-100	Version 2.4 of Dave Wecker's terminal emulator, with Xmodem and Kermit file transfer protocols

Fred Fish Disk 48

Bru	Alpha version of a hard disk file archiver
Comm	Version 1.30 of a terminal emulator with phone directories
Csh	Version 2.04 of Matt Dillion's Unix 'csh'-like CLI replacement, including Lattice & Manx C source
Diskperf	Disk benchmark program for Unix and Amiga
Du	Computes disk storage of a file or directory
MemWatch	Program to watch for programs that trash low memory. It attempts to repair the damage, and puts up a requester to inform you of the damage. From the Software Distillery.
Profiler	A realtime execution profiler for Manx C programs. Includes C source.

Fred Fish Disk 49

Cycloids	Update of electronic spirograph from disk 27
DirUtil	Enhanced version of DirUtil from disk 35
MultiDef	Scans a set of object modules and libraries searching for multiply defined symbols
MyUpdate	Disk update utility with options for stripping comments from C header files, and interactive verification of the updating process
Plot	Computes and displays 3 dimensional functions in hires
Polygon	Moire type pattern generator with color cycling
QMouse	Queries whether a mouse button is pressed. This can give a return code that can customize a startup-sequence based on whether a mouse button was pressed.
Touch	Example of setting the datestamp on a fiile, using a technique from Commodore-Amiga
Trees	More extensive version of the trees program on Disk 31

Fred Fish Disk 50

Asm	Version 1.1 of a shareware 68000 macro assembler, compatible with the Metacomco assembler. This includes an example startup module and more Motorola mneumonics.
BreakOut	A brick breakout game, uses 3-D glasses
DiskZap	Version 1.1 of a program to edit disks and binary files
FirstSilicon	A smart CLI replacement with full editing and recall of previous commands
Missile	A Missile Command-type game, with sound, in assembler
PerfectSound	Sound editor for a low-cost sound digitizer
Sizzlers	Graphics demos
UnixArc	Ver of 'arc' for Unix System V machines,in C
Wombat	Version 3.01 of Dave Warker's terminal emulator

Fred Fish Disk 51

Bison	GNU for Unix 'yacc', working update to FF4
Compress	Update to the file compression program on Disk 6
Cos	"Wheel of Fortune"-type game in AmigaBasic
DifSsed	Unix-like 'dif' and 'ssed' for finding the differences between two files, and then recreating the other, given one file, and the list of differences.
Sq, Usq	Portable versions of the CP/M squeeze and unsqueeze

Fred Fish Disk 52

Assign	Replacement for AmigaDOS 'assign' command in C
Fractal	Makes random fractal terrains
Poly, HAMPoly	Workbench-type demos for making polygons in lores and HAM
MxGads	Example of mutual exclusion gadgets with GadgetText
Tek4010	Tektronix 4010 terminal emulator
VDraw	Versions 1.16 and 1.19 of a Deluxe Paint-like drawing program

Fred Fish Disk 53

Animations	Demo animations with player program for Aegis Animator
ARCre	Creates rename scripts for files with long names, so they can be easily 'arc'ed and un'arc'ed.
ARP	Preliminary AmigaDOS replacements for 'break', 'cd', 'chmod', 'echo', 'filenote' and 'makedir'
Compiler	Not fully ported to the Amiga, this is a 68000 C compiler. It wil produce simple assembly language output, but needs a lot of work.
Spreadsheet	Update with source of the 'vc' spreadsheet on disk 36
TarSplit	Port of program to split Unix 'tar'archives
UUencode	Utilties to encode and decode binary files for ASCII transmission, expanding them by 35%

Fred Fish Disk 54

Hanoi	Solves Towers of Hanoi Problem in it's own Workbench window. by Ali Ozer
ISpell	Port of a Unix screen oriented, interactive spelling checker. (Expansion RAM required) by Pace Willisson
Ing	A Screen of lots of bouncing little windows by Leo 'Bols Ewhac' Schwab
Lav	Displays number of tasks in run queue, averaged over last 1, 5, and 15 minute periods. by Willaim Rucklidge
MIDITools	Programs to play/record through the MIDI I/F. by Fred Cassirer
MoreRows	Program to make the Work Bench Screen larger than normal. by Neil Katin and Jim Mackraz
Tilt	Program to make your Amiga look like it didn't pass vibration testing. by Leo ' Bols Ewhac' Schwab

Fred Fish Disk 55

Csh	V2.05 of Matt Dillon's csh like shell (Modified for Manx C). by Matt Dillon, Modified by Steve Drew
NewStartups	New C Startup modules:
AStartup.asm	with 1.2 fixes and better quote handling.
TWStartup.asm	opens a stdio window, using user specs. by Commodore, posted to BIX by Carolyn Schepper
Palette	Change another program's screen colors. by Carolyn Schepper
PipeDevice	Allows the standard output of one process to be fed to the standard input of another. by Matt Dillon
ScreenSave	Save a normal or HAM mode screen as an IFF file. by Carolyn Schepper
ShanghaiDemo	Demo of the Activision game Shanghai.
SoundExample	A double buffered sound example for Manx C. by Jim Goodnow
Vsprites	A working vsprite example. by Eric Cotton
Vt100	V2.6 of Dave's Vt100 terminal emulator with kermit and xmodem. by Dave Wecker

Fred Fish Disk 56

ClipBoard	Clipboard device interface routines, to provide a standard interface. by Andy Finkle
ConPackets	Demos the use of DOS Packets, ConUnit, etc. by Carolyn Schepper
GetDisks	Program to find all available disk device names and return them as an exec list. by Philip Lindsay
GetVolume	Program to get volume name of the volume that a given file resides on. by Chuck McManis
Icon2C	Reads an icon file and writes out a fragment of C code with the icon data structures. by Carolyn Schepper
MergeMem	Program to merge the MemList entries of sequentially configured RAM boards. by Carolyn Schepper
mCAD	An object oriented drawing program, V1.1 by Tim Mooney

Fred Fish Disk 57

Replaced by FF97 Due to Copyright problems

Fred Fish Disk 58

ASDG-rrd	Extremely useful shareware recoverable ram disk. by Perry Kivolowitz
BigView	Displays any IFF picture, independent of the physical display size, using hardware scroll. by John Hodgson
EGraph	Reads pairs of x and y value from a list of files and draws a formatted graph. by Laurenece Turner
HyperBase	Shareware data management system. V1.5
MemClear	Walks through the free memory lists, zeroing free memory along the way. by John Hodgson
NewZAP	A third-generation multi-purpose file sector editing utility. V3.0 by John Hodgeson
RainBow	A Maurauder-Style rainbow generator. by John Hodgson
SMUSPlayers	Two SMUS plays, to play SMUS IFF music formatted files. by John Hodgson
View	A tiny ILBM viewer by John Hodgson
WBdump	JX-80 optimized workbench printer that does not use DumpRPort. by J. Hodgson

Fed Fish Disk 59

Browser	Update to browser program on disks 18 and 34. S-E
Browser2	Another different browser program. E
Clock	Clock program with fonts, colors. E
Dme	Dillon text editor V1.22 for programmers,ED
DropCloth	Puts pattern on Workbench backdrop,ED
DropShadow	Puts shadows on Workbench windows,E-D
FixWB	Similar to DropCloth, but doesn't work yet. S-D
mCAD	Object-oriented drawing program, version 1.2.2. Much improved over disk 56.
Robotroff	Demo of animated pointers on Workbench. S-E-D
Supermort	General compounding/amortization loan calculator. E-D

Fred Fish Disk 60

Various shareware and freeware programs

Blitz	Memory resident file viewer. Very fast. E-D
BlitzFonts	Makes text output faster. E-D
HandShake	Terminal emulator with VT52/VT100/ VT102support. E-D
Med	Mouse-driven text editor version 2.1. E-D
PrtDrvGen	Generates printer drivers, version 1.1.S available from author. E-D
Show	Slideshow-like IFF viewer, V2.1. E-D
Uedit	Customizable text editor V2.0. E-D
Ueturbo	Example Uedit setup macros. S-E-D

Fred Fish Disk 61

ATPatch	Patches Transformer to work under AmigaDOS 1.2. S-E-D
FillDisk	Writes zeroes to free blocks on a disk for security. S-E-D
LPatch	Patch for programs that abort when loading under AmigaDOS 1.2. S-E-D
MicroEmacs	Conroy MicroEmacs V3.8b, newer than disk 22. S-E-D
PearlFont	Like Topaz, but rounded edges.
Terrain	Generates fractal scenery. S-E-D
VSprites	Makes 28 Vsprites, from P&E Book.

Fred Fish Disk 62

This is a port of the Unix game 'Hack', by the Software Distillery, version 1.0.3D.

Fred Fish Disk 63

This is a port of the Unix game 'Larn', by the Software Distillery, version 12.0B.

Fred Fish Disk 64

This is an offical IFF specification disk from Commodore, an update to disk 16.

Fred Fish Disk 65

Bawk	Unix text processor, like 'awk'. Doesn't work, but source is included. S-E-D.
MWB	Example of rerouting Workbench window open calls to another custom screen. Version 1.01, S-E-D
CloseWB	Example for closing a custom Workbench screen. S-E-D
Cookie	Generates one-line fortune-cookie aphorisms. S-E-D
JTime	Build-your-own mouse port clock.
MenuBuilder	Creates C source files for menus, based on text descriptions. S-E-D.
NewPackets	CBM tutorial on new packets and structures in AmigaDos 1.2.
PascalToC	Pascal to C translator, not so great. S-E-D
Prep	'ratfor'-like FORTRAN preprocessor. S-E-D
RunBack	Starts programs from CLI, allowing CLI window to close. E-D

HO! HO! HO!

Subscribe TODAY and save over 49% on Amazing Computing!

Amazing Computing is available to United States subscribers at $24.00 for 12 issues. That's a savings of $23.40 off the newsstand price. Canada and Mexico, $36.00. Foreign Surface to all other countries, $44.00. (Air mail and first class rates available upon request.)

Enter my 12 issue subscription today!
Name____
Street____
City ______________________ St.____ Zip __________

Amount Enclosed ____________

Sorry, we cannot accept credit cards or requests for billing at this time.
All funds must be in U.S. Currency drawn on a U.S. Bank.
Please allow four to six weeks for processing.

Please Mail to:

Amazing
COMPUTING
P.O. Box 869
Fall River, MA 02722-0869

Place this order form in an envelope with your Check or Money Order.

The Perfect Gift!

I will subscribe TODAY and save over 49% on Amazing Computing, plus receive my Amazing Product Guide: Winter '89.

I can't pass up an offer like this! Enclosed is my check for $24 for a one-year subscription to Amazing Computing. Canada and Mexico, $36.00. Foreign Surface to all other countries, $44.00.
(Air mail and first class rates available upon request.)

Enter my 12 issue subscription today!

Name____

Street____

City ______________________ St.____ Zip __________

Amount Enclosed ____________

Sorry, we cannot accept credit cards or requests for billing at this time.
All funds must be in U.S. Currency drawn on a U.S. Bank.
Please allow four to six weeks for processing. We are not responsible for late enteries.

Please Mail to:

Amazing
COMPUTING
P.O. Box 869
Fall River, MA 02722-0869

Place this order form in an envelope with your Check or Money Order.

VOTE

Your Civic AMIGA Duty Calls!

What are the best software packages for the Amiga?

Don't ask us.

It's all up to you.

At AmiExpo/New York in March, **YOU** will present awards to the Amiga's top software developers. Sure, we could pick out our favorites, but no one is more qualified than you, the users. You use Amiga software everyday to solve problems, explore creativity, and just have fun. *Amazing 's* job is just to report what you are doing with the Amiga!

And the best part is you don't have to break through crowds of hand-shaking poll hawkers. You don't even have to be at AmiExpo! Just complete the ballot at the bottom of this page and mail it to *Amazing* by February 15, 1989. *Amazing Computing* will tally up your choices and hand out awards directly from your votes.

At AmiExpo/Los Angeles last October, *Amazing Computing* announced awards to individuals who have dedicated their hard work and creative power to the Amiga community. Now we leave it up to you, active members of that same Amiga community, to choose the award-winning products. *Amazing* recognizes individuals, but only you can recognize the most useful products.

What are those priceless programs that bring the power of your Amiga to life and make life in general a whole lot easier? Now rave about your favorites. Stand up and be counted. Cast your Amiga ballot.

STAND UP AND BE COUNTED!

Which Amiga Products Do You Think Are Award-Winners?

Cast your Amiga Ballot now by filling in your choice for the following questions.

Name ____________________
Address ____________________
City____________________
State __________ Zip __________

SOFTWARE

Choose your preferred:

WORD PROCESSOR:____________________
DATABASE:____________________
SPREADSHEET:____________________
DESKTOP PUBLISHING package:____________________
GRAPHICS/PAINT package:____________________
CAD/DRAW package:____________________
ACCOUNTING package:____________________
TELECOMMUNICATIONS software:____________________
____
PROGRAMMING LANGUAGE:____________________
PROGRAMMING UTILITY:____________________
FILE UTILITY:____________________
ANIMATION/VIDEO software:____________________
GAME:____________________
AUDIO/MUSIC software:____________________
EDUCATIONAL software:____________________
BUSINESS software:____________________

HARDWARE

Choose your preferred:

MASS STORAGE device:____________________
ACCELERATOR:____________________
DIGITIZER/FRAME GRABBER:____________________
PRINTER:____________________
MONITOR:____________________
INPUT device:____________________
MODEM:____________________

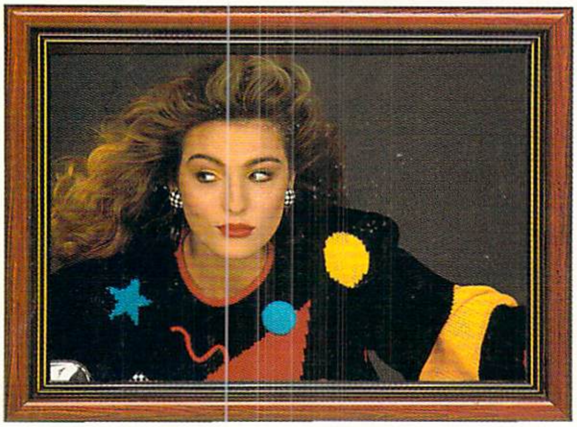
Overall, the **MOST USEFUL** Amiga product in my library is:____________________

Please return to:
Reader Ballots Amazing Computing
P.O. Box 869 • Fall River, MA 02722

DIGI-VIEW GOLD

All-New! Hardware and Software Designed for A500/2000— Even Better Pictures Than Before!

1.



2.

Actual 4096 color Digi-View Gold picture

3. Simply the Best.

The all new Digi-View Gold is the best video digitizer for the Amiga. Period. Nothing else even comes close. Why? The secret is that Digi-View Gold captures 2.1 million colors in memory, giving you an incredible 100,000 apparent colors on screen simultaneously.

And it's easy to use. Just focus your video camera on any object or picture, and in seconds Digi-View Gold turns it into Amiga graphics that glow with vibrant color and clarity. Whether you are creating graphics for desktop publishing, presentations, video, or just for fun, Digi-View Gold gives you dazzling images with amazing simplicity.

Digi-View Gold is designed specifically for the Amiga 500 and 2000, and plugs directly into the parallel port. Digi-View Gold's powerful image capture and manipulation software (version 3.0) now has complete control of color and sharpness, full overscan, extra halfbrite, and a special line art mode for desktop publishing.

Only Digi-View Gold:

- Can digitize in all Amiga resolution modes from 320x200 up to 768x480 (full hi-res overscan)
- Uses 2 to 4096 colors (including extra halfbrite)
- Uses exclusive Enhanced HAM for super fine detail
- Is 100% IFF compatible and works with any graphics software
- Can digitize 21 bits per pixel (2.1 million colors) for the highest quality images possible
- Has advanced dithering routines that give an apparent 100,000 colors on screen simultaneously
- Has powerful Image processing controls for complete IFF picture manipulation

If you want the highest quality graphics for your Amiga, as easy as 1, 2, 3; then you need the new version of the best selling video digitizer of all time: Digi-View Gold.

Only $199.95

Digi-View Gold is available now
at your local Amiga dealer.
Or call 1-800-843-8934

NEWTEK
INCORPORATED

*Requires standard gender changer for use with Amiga 1000. Video camera required; not included. NewTek sells a video camera, copy stand, and the Digi-Droid automated filter wheel for Digi-View Gold. If your local retailer doesn't carry these products, call us at 913-354-1146. Digi-View Gold is a trademark of NewTek, Inc. Amiga is a trademark of Commodore-Amiga, Inc. Be seeing you!!